

Table of Contents

M-Graphics User's Manual

Introduction

Chapter 1	Using This Manual	1-1
	<i>Introduction.....</i>	<i>1-1</i>
	<i>Key Concepts.....</i>	<i>1-2</i>
	Chapter Organization	1-2
	Online Help	1-2
	Manual Conventions	1-2
Chapter 2	Introduction to M-Graphics	2-1
	<i>Introduction.....</i>	<i>2-1</i>
	<i>Key Concepts.....</i>	<i>2-2</i>
	M-Graphics Hardware and Software Requirements.....	2-2
	M-Graphics Features	2-2
	M-Graphics File Name Extensions.....	2-2
Chapter 3	Getting Started	3-1
	<i>Introduction.....</i>	<i>3-1</i>
	<i>Key Concepts.....</i>	<i>3-2</i>
	M-Graphics Screen	3-2
	Work Area	3-2
	Toolbars	3-3
	M-Graphics Version-Checking Feature	3-6
	Menus	3-7
	Mouse Functions.....	3-15
	Runtime Mode	3-15
	Runtime Menus	3-16
	<i>Procedure Overview.....</i>	<i>3-18</i>
	<i>Detailed Procedures</i>	<i>3-19</i>
	Starting M-Graphics	3-19
	Creating a New Display.....	3-19
	Opening a Display	3-19

Saving a Display	3-19
Printing a Display	3-19
Enabling the M-Graphics Version-Checking Feature	3-20
Using Zoom	3-20
Managing the Display Area	3-21
Defining Summary Information	3-21
Viewing Statistics	3-21
Viewing Object Count Statistics	3-22
Starting Runtime	3-22
Loading a Display in Runtime	3-23
Printing the Full Screen in Runtime	3-23
Exiting M-Graphics	3-23
 Chapter 4 Customizing M-Graphics	4-1
<i>Introduction</i>	4-1
<i>Key Concepts</i>	4-2
M-Graphics Custom Settings	4-2
Command Line Options	4-2
Application Preferences	4-2
Display Properties	4-3
Display Configuration Mode Password	4-3
<i>Procedure Overview</i>	4-4
<i>Detailed Procedures</i>	4-5
Customizing the Launch of M-Graphics and Windows	4-5
Selecting a Language	4-6
Defining General Application Preferences	4-7
Defining Window Application Preferences	4-8
Defining Runtime Application Preferences	4-10
Defining Grid Application Preferences	4-12
Defining Load Tabs Application Preferences	4-14
Defining ToolTips Application Preferences	4-15
Copying Application Preferences to Display Properties	4-17
Saving Display Properties as Application Preferences	4-17
Resetting Default Application Preferences and Display Properties	4-17
Defining a Configuration Mode Password	4-17
Setting Up the Alarm Icon Path for ActiveX® Control	4-19
 Chapter 5 Creating Objects	5-1
<i>Introduction</i>	5-1
<i>Key Concepts</i>	5-2
Configure Mode	5-2
Draw Functions	5-2
M-Graphics Symbols	5-2
Insert Object	5-2

Color Palette	5-2
Color Dialog Box	5-3
Wildcards in Find Dialog Box	5-4
Select Handle Color Definitions	5-4
Text Rotation	5-4
Procedure Overview	5-5
Detailed Procedures	5-6
Drawing a Line	5-6
Drawing a Segmented Line	5-6
Drawing a Rectangle/Square	5-6
Drawing an Ellipse/Circle	5-6
Inserting Text	5-6
Defining Text Font	5-7
Drawing an Arc	5-7
Defining Line Style	5-7
Defining Line Width	5-7
Filling an Object with Color	5-7
Importing Bitmaps	5-8
Importing Metafiles	5-8
Importing Symbols	5-8
Exporting a Metafile	5-8
Freezing an Object	5-8
Inserting an Object	5-9
Troubleshooting	5-10
Common Problems	5-10
Chapter 6 Editing Objects	6-1
Introduction	6-1
Key Concepts	6-2
Edit Functions	6-2
Paste vs. Paste Special	6-2
Find vs. Replace Functions	6-2
Procedure Overview	6-3
Detailed Procedures	6-4
Editing the Length of a Line	6-4
Repositioning a Line	6-4
Editing the Vertices of a Segmented Line	6-4
Resizing a Rectangle	6-4
Resizing an Ellipse	6-5
Resizing an Arc	6-5
Editing the Shape of an Arc	6-5
Cutting an Object	6-5
Copying an Object	6-5

Pasting an Object.....	6-6
Using Paste Special.....	6-6
Duplicating an Object.....	6-6
Finding an Object.....	6-7
Replacing an Object.....	6-9
Reporting an Object.....	6-10
Chapter 7 Arranging Objects.....	7-1
Introduction.....	7-1
Key Concepts.....	7-2
Layers	7-2
Procedure Overview.....	7-4
Detailed Procedures	7-5
Grouping Objects	7-5
Ungrouping Objects	7-5
Repositioning Objects	7-5
Rotating Objects.....	7-5
Flipping Objects	7-5
Aligning Objects	7-6
Spacing Objects Evenly	7-6
Making Objects the Same Size	7-6
Adding Layers	7-7
Removing Layers	7-8
Duplicating Layers.....	7-9
Editing Layer Properties.....	7-10
Setting Currently Active Layer.....	7-11
Chapter 8 Using Templates	8-1
Introduction.....	8-1
Key Concepts.....	8-2
Templates	8-2
Template Object.....	8-2
Procedure Overview.....	8-3
Detailed Procedures	8-4
Creating a Template.....	8-4
Inserting Template Objects	8-4
Applying a Template	8-4
Removing an Applied Template	8-4
Editing an Applied Template	8-5
Updating Template Displays	8-5
Chapter 9 Using the Property Inspector	9-1

Introduction.....	9-1
Key Concepts.....	9-2
Property Inspector.....	9-2
Update Shared Objects.....	9-3
Procedure Overview.....	9-4
Detailed Procedures.....	9-5
Editing Object Parameters in the Property Inspector.....	9-5
Updating Shared Objects.....	9-5
Chapter 10 Establishing Data Source Connections.....	10-1
Introduction.....	10-1
Key Concepts.....	10-2
Data Source Connections.....	10-2
OPC Universal Tag Browser.....	10-3
Tags.....	10-5
Constant Values.....	10-5
Local M-Graphics Variables.....	10-6
State Fields.....	10-6
Simulated Values.....	10-7
Expressions.....	10-8
Procedure Overview.....	10-11
Detailed Procedures.....	10-12
Connecting to a Data Source.....	10-12
Operating the OPC Universal Tag Browser.....	10-14
Adding a Simulated Variable.....	10-16
Editing Expressions.....	10-17
Editing Local Variables.....	10-18
Creating an Alias.....	10-19
Editing an Alias.....	10-19
Setting Runtime Alias.....	10-20
Chapter 11 Adding Dynamics.....	11-1
Introduction.....	11-1
Key Concepts.....	11-2
Gradient Fill Feature.....	11-2
Gradient Styles.....	11-3
Dynamics.....	11-4
Action Dynamics.....	11-4
Selector Dynamics.....	11-8
Intrinsic Dynamics.....	11-9

Procedure Overview 11-11**Detailed Procedures 11-12**

Adding a Size Dynamic	11-12
Adding a Location Dynamic	11-14
Adding a Rotation Dynamic.....	11-16
Adding a Hide Dynamic.....	11-18
Adding a Color Dynamic.....	11-20
Adding an Analog Color Dynamic	11-22
Adding a Flash Dynamic	11-24
Adding a Pick Dynamic	11-26
Adding a Digital Selector Dynamic	11-28
Adding an Analog Selector Dynamic.....	11-30
Adding an Animator Dynamic.....	11-32
Adding a Time/Date	11-34
Creating a Process Point/Data Entry (PPT/DE)	11-35
Creating a Pushbutton	11-37
Creating a Checkbox.....	11-38
Creating a Radio Button	11-40
Creating a Display Button	11-42

Chapter 12 Using Visual Basic for Applications (VBA)..... 12-1**Introduction..... 12-1****Key Concepts..... 12-2**

VBA Features	12-2
Windows.....	12-2
Events	12-2
Object Oriented Programming.....	12-3
Development Using Visual Basic for Applications.....	12-3
VBA Editor.....	12-4
Project	12-5
M-Graphics VBA Project	12-8
M-Graphics Native Display Objects.....	12-10
Unique Object Names	12-12
Additional Information	12-12
VBA Wizards	12-13
VBA Wizard Creation Tool - Macro	12-15

Procedure Overview..... 12-16**Detailed Procedures 12-17**

Creating and Editing VBA Scripts	12-17
Adding Event Handlers	12-17
Displaying the <i>ThisDisplay</i> Module Code.....	12-18
Creating a Macro.....	12-19
Editing a Macro	12-21
Stepping a Macro	12-21
Running a Macro.....	12-22

Cleaning VBA Unused Modules	12-22
Examples	12-22
Troubleshooting	12-29
 Chapter 13 M-Graphics ActiveX Control	13-1
Introduction.....	13-1
Key Concepts.....	13-2
M-Graphics ActiveX Control	13-2
Runtime Window Properties Mode.....	13-2
Procedure Overview.....	13-3
Detailed Procedures	13-4
Inserting M-Graphics ActiveX Control	13-4
Configuring M-Graphics ActiveX Control.....	13-5
 Chapter 14 TWXView32 ActiveX Control	14-1
Introduction.....	14-1
Key Concepts.....	14-2
TWXView32 ActiveX Control.....	14-2
TWXView32 ActiveX in M-Graphics	14-3
Connections	14-3
TWXView32 in Configure and Runtime Modes.....	14-4
Runtime TWXView32 Toolbar	14-4
International Language Support.....	14-5
Asynchronous Downloading.....	14-5
Internal Architecture of ActiveX.....	14-5
TWXView32 Object Capability	14-5
TWXView32 Properties	14-6
OLE Automation.....	14-8
Methods/Properties for the TWXView32 ActiveX	14-8
Methods/Properties for Supporting Objects	14-11
Procedure Overview.....	14-15
Detailed Procedures	14-16
Inserting a TWXView32 Control	14-16
Editing General Parameters	14-18
Editing Fonts Parameters.....	14-20
Editing Workspace Parameters.....	14-21
Editing Grids Parameters	14-22
Editing Ranges Parameters	14-23
Editing Details Parameters.....	14-25
Editing Time Parameters.....	14-26
Editing Rate Parameters	14-27

Editing Pens Parameters	14-28
Editing Settings Parameters	14-29
Viewing About Information	14-30
Editing Pens in Runtime	14-30
Editing Trend in Runtime	14-32
Editing Period in Runtime	14-33
Toggling Freeze/Unfreeze in Runtime	14-33
Viewing Statistics in Runtime	14-34
Editing the Focus Pen in Runtime	14-34
Moving Between Samples in Runtime	14-35
 Chapter 15 GWXGauge ActiveX Control	15-1
Introduction	15-1
Key Concepts	15-2
GWXGauge ActiveX Control	15-2
GWXGauge ActiveX Properties	15-2
Procedure Overview	15-4
Detailed Procedures	15-5
Inserting a GWXGauge ActiveX Control	15-5
Editing General Parameters	15-7
Editing Background Parameters	15-9
Editing Caption Parameters	15-10
Editing Bar Parameters	15-11
Editing Needle Parameters	15-12
Editing Value Parameters	15-13
Editing Scale1 Parameters	15-14
Editing Scale2 Parameters	15-15
Editing Fonts Parameters	15-16
Editing Slider/Dial Parameters	15-17
Editing Warning Zones Parameters	15-19
Viewing the About Information	15-20
 Chapter 16 GWXSwitch ActiveX Control	16-1
Introduction	16-1
Key Concepts	16-2
GWXSwitch ActiveX Control	16-2
GWXSwitch ActiveX Properties	16-2
Procedure Overview	16-4
Detailed Procedures	16-5
Inserting a GWXSwitch ActiveX Control	16-5
Editing General Parameters	16-7
Editing Background Parameters	16-9

Editing Caption Parameters	16-10
Editing Labels Parameters	16-11
Editing Fonts Parameters.....	16-12
Viewing the About Information	16-13
Chapter 17 GWXSlider ActiveX Control.....	17-1
<i>Introduction.....</i>	<i>17-1</i>
<i>Key Concepts.....</i>	<i>17-2</i>
GWXSlider ActiveX Control.....	17-2
GWXSlider ActiveX Properties	17-2
<i>Procedure Overview</i>	<i>17-4</i>
<i>Detailed Procedures</i>	<i>17-5</i>
Inserting a GWXSlider ActiveX Control	17-5
Editing General Parameters	17-7
Editing Slider Parameters	17-9
Editing Slow Spin Buttons Parameters.....	17-11
Editing Fast Spin Buttons Parameters	17-12
Editing Scale Parameters.....	17-13
Editing Background Parameters	17-15
Editing Caption Parameters	17-16
Editing Value Parameters	17-17
Editing Fonts Parameters.....	17-18
Viewing the About Information	17-19
Chapter 18 AWXView32 ActiveX® Control.....	18-1
<i>Introduction.....</i>	<i>18-1</i>
Appendix A OLE Automation Reference	A-1
<i>Introduction.....</i>	<i>A-1</i>
<i>Key Concepts.....</i>	<i>A-2</i>
Properties and Methods	A-2
Custom Configuration	A-2
Custom Command Execution.....	A-3
GwxDisplay	A-3
GwxVisible	A-17
GwxText	A-19
GwxRectangle	A-20
GwxArc.....	A-20
GwxSymbol	A-21
GwxOleObject	A-21
GwxEllipse	A-21
GwxLine	A-21
GwxBitmap	A-21

GwxMetafile.....	A-21
GwxButton.....	A-22
GwxDynamic	A-22
GwxDigitalSelector.....	A-23
GwxDigitalSelectorInfo.....	A-23
GwxDigitalColor.....	A-24
GwxDigitalColorInfo.....	A-24
GwxAnalogSelector.....	A-24
GwxAnalogColor	A-24
GwxAnimator.....	A-24
GwxSize	A-24
GwxLocation	A-24
GwxRotation.....	A-24
GwxFlash	A-24
GwxHide.....	A-24
GwxPick	A-24
GwxProcessPoint.....	A-25
GwxTimedate	A-25
GwxPoint.....	A-25
GWXview32.....	A-26

Appendix B **DIN Symbol Library..... B-1**

Introduction..... B-1

Key Concepts..... B-2

Symbol Library B-2

DIN Symbols Examples..... B-3

Appendix C **French Symbol Library C-1**

Introduction..... C-1

Key Concepts..... C-2

Symbol Library C-2

French Symbols Examples..... C-3

Appendix D **Johnson Controls Symbol Library D-1**

Introduction..... D-1

Key Concepts..... D-2

Symbol Library D-2

Johnson Controls Symbols Examples..... D-3

Appendix E **Other Symbol Library E-1**

Introduction..... E-1

Key Concepts..... E-2

Symbol Library	E-2
Other Symbols Examples	E-3
Appendix F Tips and Hints	F-1
<i>Introduction.....</i>	<i>F-1</i>
<i>Key Concepts.....</i>	<i>F-2</i>
Characteristics of Good Graphics	F-2
Usable Graphics.....	F-3
Keyboard Shortcuts.....	F-5
Additional M-Graphics Shortcuts and Features	F-6
Graphic Libraries.....	F-6
<i>Procedure Overview.....</i>	<i>F-7</i>
<i>Detailed Procedures</i>	<i>F-8</i>
Adding Graphic Hot Links to Other Screens	F-8
Adding Fan Rotation to a Graphic.....	F-8
Adding Command Operation Box to Object when Clicking Left or Right Mouse Button	F-9
Adding Focus Window Screen to Object when Clicking Left or Right Mouse Button.	F-9
Creating a Slider Bar	F-10
Adding On/Off Binary Commands to Pushbuttons	F-10
Adding Override Binary Commands to Pushbuttons.....	F-11
Adding the Auto Binary Command to a Pushbutton.....	F-11
Appendix G Guidelines and Recommendations for Better Graphics	
Performance with Integrated N1 Networks	G-1
<i>Introduction.....</i>	<i>G-1</i>
<i>Key Concepts.....</i>	<i>G-2</i>
Guidelines and Recommendations for Better Graphics Performance Overview.....	G-2
Modifiable Parameters	G-3
Maximum Update Rate Default Value	G-4
Performance Tips.....	G-4
<i>Detailed Procedures</i>	<i>G-6</i>
Changing the Maximum Update Rate	G-6
Selecting Multiple Files and Changing the Maximum Update Rate	G-7

Chapter 1

Using This Manual

Introduction

M-Graphics Release 4.0 is a fully compliant OLE (Object Linking and Embedding) for Process Control (OPC) software package featuring ActiveX® and OLE technologies. Your creativity is the cornerstone to the dynamic visual impact that is provided with M-Graphics. M-Graphics encourages system utilization, enhances productivity, and lowers overall operating costs.

Note: M-Graphics Release 4.0 files are not backward compatible with Release 3.0 or older revisions.

Key Concepts

Chapter Organization

Each chapter in the *M-Graphics User's Manual* is divided into five main sections:

Table 1-1: Chapter Organization

Section	Description
<i>Introduction</i>	Briefly outlines the contents of the chapter.
<i>Key Concepts</i>	Describes background information necessary to perform or understand specific tasks.
<i>Procedure Overview</i>	Describes general steps for performing the tasks which are described in detail in the <i>Detailed Procedures</i> section. This section is geared toward users already experienced in using M-Graphics who only need a reference.
<i>Detailed Procedures</i>	Describes in detail the steps needed to complete specific tasks described within the chapter. This section is geared toward users who are new to M-Graphics.
<i>Troubleshooting</i>	Provides information on potential problems as well as methods for solving them.

If a section is not necessary for a particular chapter, it is omitted. For example, this chapter, *Using This Manual*, does not contain a *Troubleshooting* section.

Online Help

M-Graphics software includes online help. The Help screens may be accessed by selecting **H**elp from the menu bar, by pressing the F1 key, or by pressing Alt+H. Help describes the basic procedures for performing functions within the M-Graphics software. The online help system follows the basic conventions of Windows® 98 and Windows NT® help systems.

Manual Conventions

As in most Windows based programs, sometimes there are multiple ways to perform the same tasks. This manual does not describe how to use both the mouse and the keyboard to perform the same step. In general, Microsoft® conventions are used to describe software features.

Chapter 2

Introduction to M-Graphics

Introduction

M-Graphics allows monitoring, control, and analysis of facilities using dynamic color graphic displays. M-Graphics also provides graphical navigation of a facility and launching of other Microsoft® ActiveX® applications. This chapter describes:

- M-Graphics hardware and software requirements
- M-Graphics features
- M-Graphics file name extensions

Key Concepts

M-Graphics Hardware and Software Requirements

M-Graphics runs standalone or within an ActiveX document container. We recommend the following:

- 2.6 GHz Pentium® 4 processor
- Windows® 2000 Professional or Windows XP Professional operating system
- 256 MB Random Access Memory (RAM) (128 MB minimum)
- 40 GB hard drive
- 17 inch monitor with 1024 x 768 minimum resolution
- video with 2 MB memory that supports 16-bit (65,535) colors

M-Graphics Features

M-Graphics features include:

- true 32-bit design
- universal OPC connectivity
- powerful display creation tools
- containment of ActiveX controls and OLE objects
- ActiveX document server
- comprehensive OLE automation interface
- fast dynamic animation
- M-Graphics active control
- advanced symbol library

M-Graphics File Name Extensions

In M-Graphics, a file can be saved as:

- M-Graphics display file (*.gdf)
- M-Graphics template file (*.tdf)
- Symbol files (*.sdf)

Getting Started

Introduction

M-Graphics can be run as a standalone program or in an ActiveX document container. This chapter describes how to:

- start M-Graphics
- create a new display
- open a display
- save a display
- print a display
- use zoom
- manage display area
- define summary information
- view statistics
- view object count statistics
- start Runtime
- load a display in Runtime
- print the full screen in Runtime
- exit M-Graphics

Key Concepts

M-Graphics Screen

Figure 3-1 shows an example of an M-Graphics screen.

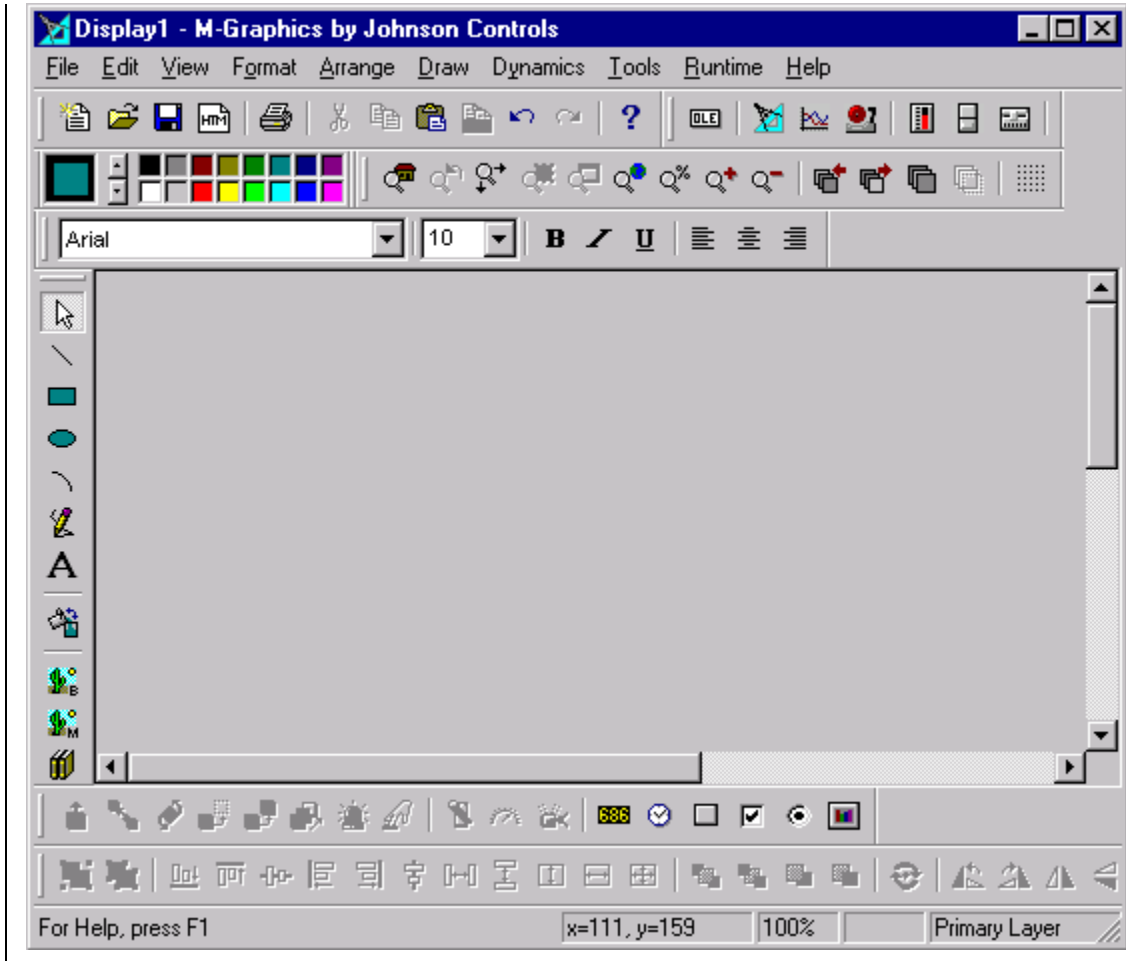


Figure 3-1: M-Graphics Screen

Work Area

The work area is where you build your M-Graphics displays.

Figure 3-1 shows only a portion of the entire work area that can be accessed by the scroll bars.

When the document is saved, information about the current view is saved with the display, so that it can be restored when the display is reloaded.

Status Bar

The status bar is displayed at the bottom of the M-Graphics screen. The left area of the status bar describes actions of menu items and toolbar buttons, as you navigate through menus or select a toolbar button. The right area of the status bar displays cursor coordinates, zoom percent, and grid snap on/off.

Toolbars

The M-Graphics tools that create displays are grouped by functionality and are available through the menu options and the toolbars.

Main Toolbar

The Main toolbar is displayed by default across the top of the M-Graphics window below the menu bar. Use the Main toolbar for basic file and display maintenance (Figure 3-2).

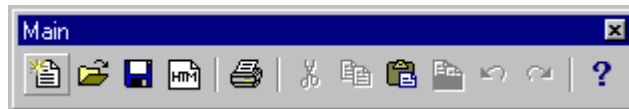


Figure 3-2: Main Toolbar

Dynamics Toolbar

Use the Dynamics toolbar to make dynamic connections to data points in OPC servers (Figure 3-3).



Figure 3-3: Dynamics Toolbar








ActiveX Toolbar

Use the ActiveX toolbar to insert objects and controls (Figure 3-4).



Figure 3-4: ActiveX Toolbar

Table 3-1: ActiveX Toolbar

ActiveX Toolbar Options		Description
OLE		Allows an ActiveX object to be inserted.
GraphWorxs		Allows a graphic to be inserted inside another graphic.
TrendGraph		Allows a TrendGraph ActiveX object to be inserted.
AlarmWorxs		Allows an AlarmWorxs ActiveX object to be inserted.
Gauge Control		Allows a Gauge ActiveX object to be inserted.
Switch Control		Allows a Switch ActiveX object to be inserted.
Slider Control		Allows a Slider ActiveX object to be inserted.

Draw Toolbar

Use the Draw toolbar for drawing, importing, and editing objects (Figure 3-5).

**Figure 3-5: Draw Toolbar****Arrange Toolbar**

Use the Arrange toolbar for grouping and ungrouping objects, for aligning objects, for rotating objects, and shifting objects in layers (Figure 3-6).



AToolbar

Figure 3-6: Arrange Toolbar

Text Style Toolbar

Use the Font toolbar for changing the font attributes of text objects in a display (Figure 3-7).



FToolbar

Figure 3-7: Text Style Toolbar

Color Palette

Use the Color Palette to select new colors during Configure mode in the M-Graphics application. The Color Palette includes fill, border, background, and text colors (Figure 3-8).

Double-clicking on the Color Palette opens the Color Palette dialog box (Figure 3-9).



CToolbar

Figure 3-8: Color Palette Toolbar



CPalette

Figure 3-9: Color Palette Dialog Box

Right-clicking on the empty area surrounding the spin button control of the Color Palette toolbar brings up Color Palette Tool menu Figure 3-10.

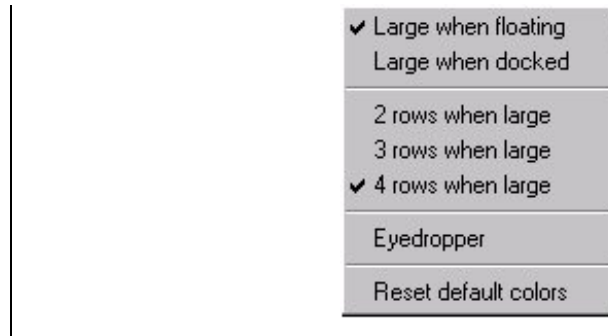


Figure 3-10: Color Palette Tool Menu

This menu can be used to change the size of the Color Palette (large/small) and to set the number of rows of colors visible at a time (2, 3, or 4). There is also an option to reset the factory default color definitions of the Color Palette.

Note: The Eyedropper feature can pick up colors from a display to add them to the Color Palette. To accomplish this, choose the Eyedropper menu item (the mouse pointer changes to an eyedropper). Click anywhere in the M-Graphics window over the color you want to pick. Deposit the color in the Color Palette by clicking over the desired color box in the rows of available colors.

Menus

The following tables describe M-Graphics menus and show corresponding toolbar buttons.

Only select menu options have corresponding toolbar buttons.

Table 3-2: File Menu





File Menu Options	Toolbar Button	Description
<u>N</u>ew		Creates a new display.
<u>O</u>pen		Opens a display.
<u>S</u>ave		Saves current display with either .tdf or .gdf extension.
Save <u>A</u>s		Saves current display with a new name or extension.
<u>P</u>rint		Prints current display.
Print <u>P</u>review		Displays file before printing.
Print <u>S</u>etup		Opens Print Setup dialog box with printer options.
Recently Used File List		Lists last four files opened in M-Graphics.
<u>E</u>xit		Closes M-Graphics.

Table 3-3: Edit Menu








Edit Menu Options	Toolbar Button	Description
<u>U</u> ndo		Reverses last editing action.
<u>R</u>edo		Repeats last editing action.
<u>C</u>ut		Cuts selected object to the clipboard.
<u>C</u>opy		Copies selected object to the clipboard.
<u>P</u>aste		Pastes object from clipboard.
<u>D</u>elete		Deletes selected object.
Paste <u>S</u>pecial		Opens Paste Special dialog box with paste options.
Duplicate		Pastes a copy of an object on work area.
Select All <u>O</u>bjects		Selects all objects.
Select All <u>D</u>ynamics		Selects all dynamics.
Delete Selected <u>D</u>ynamics		Deletes selected dynamics.
Delete All <u>D</u>ynamics		Deletes all dynamics.
<u>F</u>ind		Opens Find dialog box which searches for objects/properties.
<u>R</u>eplace		Opens Replace dialog box which replaces specified objects/properties.
<u>R</u>eport		Opens Report dialog box which generates a listing of specified objects.
Update Shared <u>O</u>bjects		Updates specified property on all shared objects.
Insert <u>N</u>ew Object		Adds a new ActiveX object to display.
<u>L</u>inks		Displays current links for an object.
<u>O</u>bject		Opens Property Inspector for selected ActiveX object.
<u>P</u>roperty Inspector		Opens Property Inspector for selected graphic object.

Table 3-4: View Menu

View Menu Options	Toolbar Button	Description
<u>H</u>ome		Returns display to last saved view of the document.
<u>U</u>nzoom		Acts as an undo for zoom.
<u>Z</u>oom		Provides 50%, 75%, 100%, 150%, 250%, and custom zoom options.
<u>B</u>ox Zoom		Allows user to drag a box around an area to zoom to.
<u>Z</u>oom Selection		Zooms to selected objects.
<u>F</u>it to Window		Sizes the display so all objects in the display fit into the window.
S<u>h</u>ow <u>W</u>hole Display		Allows user to see the entire work area.
<u>D</u>ecluttering Zoom		Zooms in one particular part of graphic.
S<u>u</u>mary <u>I</u>nfo		Opens a dialog box containing summary information and statistics.
O<u>b</u>ject <u>C</u>ount		Displays statistics for objects, actions, and display sizes.
H<u>i</u>de <u>L</u>ayers		Allows user to hide/show one or more layers on the work area.
T<u>o</u>olbars		Hides/shows toolbars.
<u>T</u>oggle Toolbars		Hides/shows toolbars.
<u>S</u>tatus Bar		Hides/shows status bar.
H<u>o</u>rzontal <u>S</u>croll Bar		Hides/shows horizontal scroll bar.
<u>V</u>ertical Scroll Bar		Hides/shows vertical scroll bar.
T<u>o</u>gg<u>l</u>e Both Scroll Bars		Hides/shows both scroll bars.
<u>G</u>rid		Hides/shows grid.
R<u>u</u>n<u>n</u>time Windows Properties Mode		Allows user to see what a display will look like in Runtime mode without entering Runtime.
<u>P</u>roperty Window		Allows user to select the properties of any ActiveX object.
S<u>e</u>lect <u>L</u>anguage		Allows user to select a language.

Table 3-5: Format Menu


Format Menu Options	Toolbar Button	Description
<u>T</u>emplate		Opens submenu with the following options: apply template, remove applied template, edit applied template, cancel edit applied template, insert template object, and update template displays.
L ayers		Opens submenu with the following options: add layer, remove layer, duplicate layer, edit layer properties, set current active layer, set active layer next, set active layer previous, hide layers above current layer, and hide layers below current layer.
S et Configuration M ode Password		Allows a password to be set for graphic configuration.
B ackground Color		Opens Color Palette for selecting a background color.
<u>F</u>ill Color		Opens Color Palette for selecting a fill color.
<u>L</u>ine Color		Opens Color Palette for selecting a line color.
L ine <u>W</u>idth		Opens submenu for selecting one of ten line widths.
L ine <u>S</u>tyl e		Opens submenu for selecting one of five line styles.
T oggle F ill		Toggles between fill and unfill.
T oggle F reeze		Toggles between freeze and unfreeze. The Freeze function disables the movement or stretching of objects in the work area.
F ont		Opens Font dialog box for selecting font, font style, size, effects, and script.
<u>D</u>isplay Properties		Shows display properties.
S ave Properties as P references		Saves display properties as application preferences.
R eset Default D isplay Properties		Resets display properties to factory default.
A pplication P references		Shows application preferences.
A pply Preferences t o Properties		Saves application preferences as display properties.
R eset Default A pplication P references		Resets application preferences to factory default.

Table 3-6: Arrange Menu
























Arrange Menu Options	Toolbar Button	Description
<u>G</u>roup into Symbol		Groups several objects into one symbol. The resulting symbol can be moved as one object.
<u>U</u>ngroup Symbol		Ungroups the symbol into individual objects.
Bring to <u>F</u>ront		Moves the selected objects on top of all the objects.
Send to <u>B</u>ack		Moves the selected objects to the bottom of all the objects in the current work area.
Bring Forward		Moves the selected objects up one layer in the display.
Send Backward		Moves the selected objects down one layer in the display.
<u>R</u>otate/Flip		Rotates the selected objects to the left in 90 degree increments.
		Rotates the selected objects to the right in 90 degree increments.
		Flips the selected objects to a horizontal position.
		Flips the selected objects to a vertical position.
Free <u>R</u>otate		Enables rotation of static objects to any angle.
<u>A</u>lign		Aligns objects according to their tops.
		Aligns objects according to their bottoms.
		Aligns objects according to their middles.
		Aligns objects according to their left sides.
		Aligns objects according to their right sides.
		Aligns objects according to their centers.
<u>S</u>pace Evenly		Aligns objects according to both centers.
		Distributes objects evenly across.
		Distributes objects evenly down.
<u>M</u>ake Same Size		Makes multiple objects in the display a uniform height.
		Makes multiple objects in the display a uniform width.
		Makes multiple objects in the display a uniform height and width.

Table 3-7: Draw Menu











Draw Menu Options	Toolbar Buttons	Description
S elect M ode		Allows selection of objects.
S traight L ine		Draws a straight line.
R ectangle/ S quare		Draws a rectangle or square.
E llipse/ C ircle		Draws an ellipse or circle.
A rc		Draws an arc.
S egmented L ine		Draws a segmented line.
T ext		Inserts text.
I mport		Inserts a bitmap.
		Inserts a metafile.
		Opens the Symbol Library.
E xport		Exports objects as a metafile.

Table 3-8: Dynamics Menu


















Dynamics Menu Options	Toolbar Button	Description
<u>A</u>ctions		Adds a Size dynamic.
		Adds a Locator/Slider dynamic.
		Adds a Rotation/Dial dynamic.
		Adds a Hide dynamic.
		Adds a Color dynamic.
		Adds an Analog Color dynamic.
		Adds a Flash dynamic.
		Adds a Pick dynamic.
<u>S</u>electors		Adds a Digital Selector dynamic.
		Adds an Analog Selector dynamic.
		Adds an Animator dynamic.
<u>I</u>ntrinsics		Creates a Process Point dynamic.
		Creates a Time/Date dynamic.
		Creates a Pushbutton dynamic.
		Creates a Checkbox dynamic.
		Creates a Radio Button dynamic.
		Activates the Button Wizard to build a Custom button.
Set <u>P</u>ick Tab Order		Determines order for Pick actions if multiple objects overlap each other.
Edit Local <u>V</u>ariables		Opens the Local Variables dialog box, so user can edit current local variables.
Edit <u>A</u>liases		Opens the Edit Aliases dialog box, so user can edit current aliases.

Table 3-9: Tools Menu

Tools Menu Options	Description
<u>M</u>acros	Opens following macro options: Create Macro, Edit Macro, Step Macro, Run Macro, Clean VBA Unused Modules, Visual Basic Editor.
S<u>e</u>t Working <u>D</u>irectory	Specifies current directory for saving graphics and templates.
<u>A</u>lias File Editor	Opens Alias File Editor dialog box to edit Runtime aliases.
E<u>x</u>port to HTML	Exports the graphic to HyperText Markup Language (HTML). As of M-Graphics Release 4.0, this feature can be used only with Iconics WebHMI™.
<u>W</u>indows CE	Specifies Windows CE preferences or Configuration for Windows.

Table 3-10: Mode Menu*

Runtime Menu Options	Description
C<u>o</u>nfigure	Switches to Configure mode.
R<u>u</u>ntime	Switches to Runtime mode.

* The Mode menu only appears when M-Graphics is in a container. If M-Graphics is running standalone, Configure or Runtime appears on the menu bar.

Table 3-11: Help Menu

Help Menu Options	Description
<u>H</u>elp Topics	Opens online help.
A<u>b</u>out Application	Opens About dialog box with copyright and version information.

Mouse Functions

Table 3-12 describes mouse functions used in M-Graphics.

Table 3-12: Mouse Functions

Mouse Function	Description
Left-click	Selects static objects.
Right-click	Opens the Format menu.
Double-click	Opens the Property Inspector.
Drag	Moves objects or selects multiple objects.
Drag Object Handles	Moves, stretches, and resizes objects.
Drag from M-Graphics to Desktop	Becomes an icon and can be dragged back as the original object.
Shift while Stretching Object	Maintains size of object in proportion.
Ctrl while Stretching Object	Resizes from center of object.
Ctrl + Drag	Copies and duplicates within the display.
Shift + Left-click	Selects/deselects multiple objects one by one.
Shift + Right-click	Edits text, arc, line, or symbol objects.

Note: If objects are selected, use cursor keys to move by one pixel.

Runtime Mode

The Runtime mode activates all bindings and dynamics. The Runtime environment functions according to how you configure the Runtime parameters in M-Graphics Configure mode.

During Runtime, you can load the screen, print the screen, close windows, exit from the application, get help, and invoke the Configure mode.

Functions **not** supported in Runtime mode:

- replacing point names (user should use delete/add, or within the VBA environment, use the Form Load event for replacing point names)
- changing of plot type of the trend
- modifying file name for saving module's data
- getting through OLE the interface to the Timesettings object
- modifying X-axis of XY plots
- modifying filename for saving statistics
- modifying preference for global ranges
- modifying preference for shifting grids
- modifying preference for startup value
- modifying preference for restoring range values

Runtime Menus

The following menus are available in Runtime:

Table 3-13: Runtime File Menu

File Menu Options	Description
<u>O</u> pen	Opens a display.
<u>P</u> rint	Prints current display.
<u>P</u> rint <u>P</u> review	Displays file before printing.
<u>P</u> rint <u>S</u> etup	Opens Print Setup dialog box with printer options.
<u>R</u> ecently <u>U</u> sed <u>F</u> ile <u>L</u> ist	Lists last four files opened in M-Graphics.
<u>E</u> xit	Closes M-Graphics.

Table 3-14: Runtime View Menu

View Menu Options	Description
<u>D</u> isplay <u>B</u> ack	Displays previously shown graphic. Shortcut is Ctrl + Pg Up.
<u>D</u> isplay <u>F</u> orward	Displays next graphic. Shortcut is Ctrl + Pg Dn.
<u>H</u> ome	Returns display to last saved view of the document.
<u>U</u> nzoom	Acts as an undo for zoom.
<u>Z</u> oom	Provides 50%, 75%, 100%, 150%, 250%, and custom zoom options.
<u>B</u> ox <u>Z</u> oom	Allows user to drag a box around an area to zoom.
<u>F</u> it to <u>W</u> indow	Sizes the display so all objects in the display fit into the window.
<u>S</u> how <u>W</u> hole <u>D</u> isplay	Allows user to see the entire work area.
<u>D</u> ecluttering <u>Z</u> oom	Provides zoom function for specific layer.
<u>H</u> ide <u>L</u> ayers	Allows user to hide layers on graphic.
<u>H</u> orizontal <u>S</u> croll <u>B</u> ar	Hides/shows horizontal scroll bar.
<u>V</u> ertical <u>S</u> croll <u>B</u> ar	Hides/shows vertical scroll bar.
<u>T</u> oggle <u>B</u> oth <u>S</u> croll <u>B</u> ars	Hides/shows both horizontal and vertical scroll bars.
<u>T</u> ool <u>T</u> ips	Hides/shows ToolTips.
<u>D</u> ata <u>S</u> tatistics	Opens the OLE Express Statistics dialog box which provides M-Graphics Runtime information.
<u>S</u> elect <u>L</u> anguage	Allows user to select a language.

Table 3-15: Tools Menu

Tool Menu Options	Description
<u>S</u> et <u>W</u> orking <u>D</u> irectory	Allows user to set up working directory.

Table 3-16: Runtime Mode Menu*

View Menu Options	Description
Configure	Switches to Configure mode.

* If M-Graphics is not running in a container, the Configure option is menu available.

Table 3-17: Runtime Help Menu

Help Menu Options	Description
<u>H</u>elp Topics	Opens online help.
About Application	Opens About dialog box with copyright and version information.

Procedure Overview

Table 3-18: Getting Started

To Do This	Follow These Steps:
Start M-Graphics	On the Windows Start menu, select Programs > Johnson Controls > M-Graphics.
Create a New Display	On the File menu, select New.
Open a Display	On the File menu, select Open. Select a display file. Click OK.
Save a Display	On the File menu, select Save or Save As. Enter a file name and file type.
Print a Display	On the File menu, select Print. Specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options. Click OK.
Use Zoom	On the View menu, select a zoom option.
Manage Display Area	On the View menu, select the options you want to view on the display. Options include toolbars, status bar, horizontal scroll bar, and vertical scroll bar.
Define Summary Information	On the File menu, select Summary Info. Select the Summary tab. Fill in the fields. Click OK.
View Statistics	On the File menu, select Summary Info. Select the Statistics tab which provides general information about the currently displayed file.
View Object Count Statistics	From the View menu, select Object Count. Click OK to return to the display.
Start Runtime	In M-Graphics, select Runtime on the menu bar.
Load a Display in Runtime	On the File menu, select Open. Select a display file and click on the OK button.
Print the Full Screen in Runtime	On the File menu, select Print. Fill in the required fields and click OK.
Exit M-Graphics	On the File menu, select Exit.

Detailed Procedures

Starting M-Graphics

To start M-Graphics:

On the Windows Start menu, select Programs > Johnson Controls > M-Graphics.

Creating a New Display

To create a new display:

On the File menu, select New.

Opening a Display

To open a display:

1. On the File menu, select Open. The Open dialog box appears.
2. Select a display file.
3. Click OK.

Saving a Display

To save a display:

1. On the File menu, select Save or Save As. The Save dialog box appears.
2. Enter a file name and file type. File types include .gdf for graphic displays, .tdf for templates, and .gdf without VBA code.

Printing a Display

To print a display:

1. On the File menu, select Print.
2. Specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options.
3. Click OK.

Using Zoom

To use zoom:

On the View menu, select a zoom option described in Table 3-4.

Managing the Display Area

To manage the display area:

On the View menu, select the options you want to view on the display. Options include toolbars, status bar, horizontal scroll bar, and vertical scroll bar and properties.

Defining Summary Information

To define summary information:

1. On the File menu, select Summary Info. The Summary Information dialog box appears.
2. Select the Summary tab.
3. Fill in the fields.
4. Click OK.

Viewing Statistics

To view statistics:

1. On the File menu, select Summary Info. The Summary Information dialog box appears.
2. Select the Statistics tab, which provides general information about the currently displayed file.

Summary Information of a display can also be viewed in Windows Explorer. In Windows Explorer, select the desired display file, right-click, and choose the Properties menu item. The Summary and Statistics tabs appear in the resulting dialog box.

Viewing Object Count Statistics

To view object count statistics:

1. From the View menu, select Object Count. The Object Count dialog box appears (Figure 3-11).

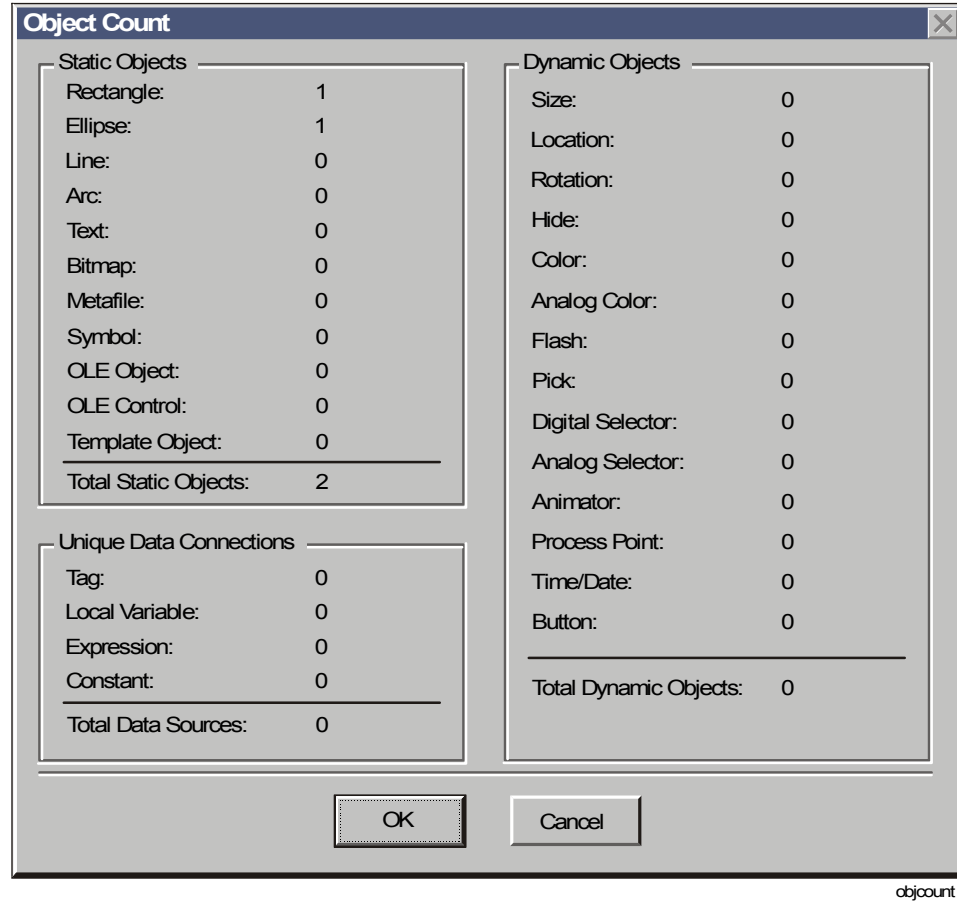


Figure 3-11: Object Count Dialog Box

2. Click OK to return to the display.

Starting Runtime

To start Runtime:

In M-Graphics, select Runtime on the menu bar.

Loading a Display in Runtime

To load a display in Runtime:

1. On the File menu, select Open. The Open File dialog box appears.
2. Select a display file and click on the OK button. The display appears in the selected window.

Printing the Full Screen in Runtime

To print the full screen in Runtime:

1. On the File menu, select Print. The Print dialog box displays.
2. Fill in the required fields and click OK.

Exiting M-Graphics

To exit M-Graphics:

On the File menu, select Exit.

IMPORTANT: If you have not saved the changes in Configure mode, and you exit during Runtime, all changes are lost.

If the M3 Workstation logs out for any reason while updating M-Graphics in a container, all changes are lost. Make sure to save all files before logging out.

Make sure to save to the correct location. The location is controlled by the container and defaults to the location of the saved .m3w file.

If you are using Metasys® Telephone Interface (MTI) and M-Graphics, and you reboot the M-Series Workstation, make sure MTI fully initializes before starting M-Graphics.

Chapter 4

Customizing M-Graphics

Introduction

This chapter explains the different ways of configuring and customizing your system. This chapter describes how to:

- customize the launch of M-Graphics and Windows
- select a language
- define general application preferences
- define window application preferences
- define Runtime application preferences
- define grid application preferences
- define load tabs application preferences
- define ToolTips application preferences
- copy application preferences to display properties
- save display properties as application preferences
- reset default application preferences and display properties
- define a Configuration mode password
- set up the alarm icon path for ActiveX® Control

Key Concepts

M-Graphics Custom Settings

Customize M-Graphics system using:

- Command line options
- M-Graphics Preference dialog box
- Display Properties dialog box

Command Line Options

The command line feature allows the user to start M-Graphics with a specified display and launch M-Graphics directly in Runtime mode.

Application Preferences

Application preferences define default operating parameters for the display editing environment. These parameters apply to the environment for **new** displays created in M-Graphics. When an existing graphic needs to have the new standards applied, the preferences can be copied to the current display properties.

General Tab

The General tab of the application preferences allows the user to define line, color, font, and general display features of the new display.

Window Tab

The Window tab of the application preferences allows the user to define the dimensions, resolution, scroll bar, and Runtime menu bar, title bar, and frame features of the new display.

Runtime Tab

The Runtime tab of the application preferences allows the user to define Runtime features of the new display.

Grid Tab

The Grid tab of the application preferences allows the user to define grid features of the new display.

Load Tabs Tab

The Load Tabs tab of the application preferences allows the user to construct a row of tabs in which each tab will load a display file. This feature allows the user to easily navigate through numerous display files.

ToolTips Tab

The ToolTips tab of the application preferences allows the user to define which text messages will pop up when the cursor is moved over an object in the new display.

Display Properties

Display properties define default operating parameters for the display editing and Runtime environments. These parameters apply to the environment for the **current** display in M-Graphics. When a graphic's properties change, save the graphic and new standards as Application Preferences.

Display Configuration Mode Password

Passwords may be attached to the configuration of an M-Graphics display. When the password is defined and entered, the Configuration mode of the display is accessible. If the incorrect password is entered, a warning message occurs and access is denied until the proper password is entered.

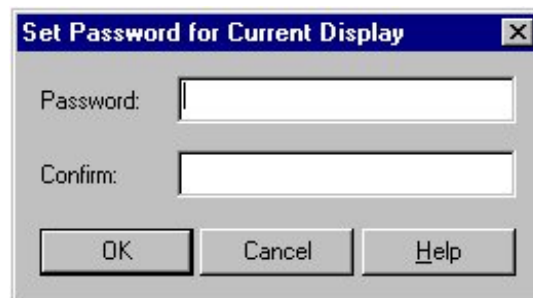


Figure 4-1: Set Password for Current Display

Note: The display password does not apply to displays opened in Runtime mode.

When trying to open a password protected display, Figure 4-2 appears before the user is able to enter the configuration.



Figure 4-2: Enter Password

Note: The challenge code is not necessary for M-Graphics login.

Procedure Overview

Table 4-1: Customizing M-Graphics

To Do This	Follow These Steps:
Customize the Launch of M-Graphics and Windows	Right-click Start and select Open. Double-click on the Programs folder. Double-click on the Johnson Controls folder. Right-click the M-Graphics icon and select Properties. Click the Shortcut tab and enter the appropriate command line options at the end of the Target field.
Select a Language	From the View menu, select Select Language. Define parameters. Click OK.
Define General Application Preferences	On the Format menu, select Application Preferences. Select the General tab. Modify the fields and click OK.
Define Window Application Preferences	On the Format menu, select Application Preferences. Select the Window tab. Modify the fields and click OK.
Define Runtime Application Preferences	On the Format menu, select Application Preferences. Select the Runtime tab. Modify the fields and click OK.
Define Grid Application Preferences	On the Format menu, select Application Preferences. Select the Grid tab. Modify the fields and click OK.
Define Load Tabs Application Preferences	On the Format menu, select Application Preferences. Select the Load Tabs tab. Modify the fields and click OK.
Define ToolTips Application Preferences	On the Format menu, select Application Preferences. Select the ToolTips tab. Modify the fields and click OK.
Copy Application Preferences to Display Properties	On the Format menu, select Apply Preferences to Properties. Select preference categories. Click OK.
Save Display Properties as Application Preferences	On the Format menu, select Save Properties as Preferences. Select properties categories. Click OK.
Reset Default Application Preferences and Display Properties	On the Format menu, select Reset Default Display Properties or Reset Default Application Preferences.
Define a Configuration Mode Password	On the Format menu, select Set Configuration Mode Password. Enter the password. Confirm the password. Click OK.

Detailed Procedures

Customizing the Launch of M-Graphics and Windows

To customize the launch of M-Graphics and Windows:

1. Right-click Start and select Open. The Start menu window appears.
2. Double-click on the Programs folder.
3. Double-click on the Johnson Controls folder.
4. Right-click the M-Graphics icon and select Properties. The M-Graphics Properties dialog box appears (Figure 4-3).

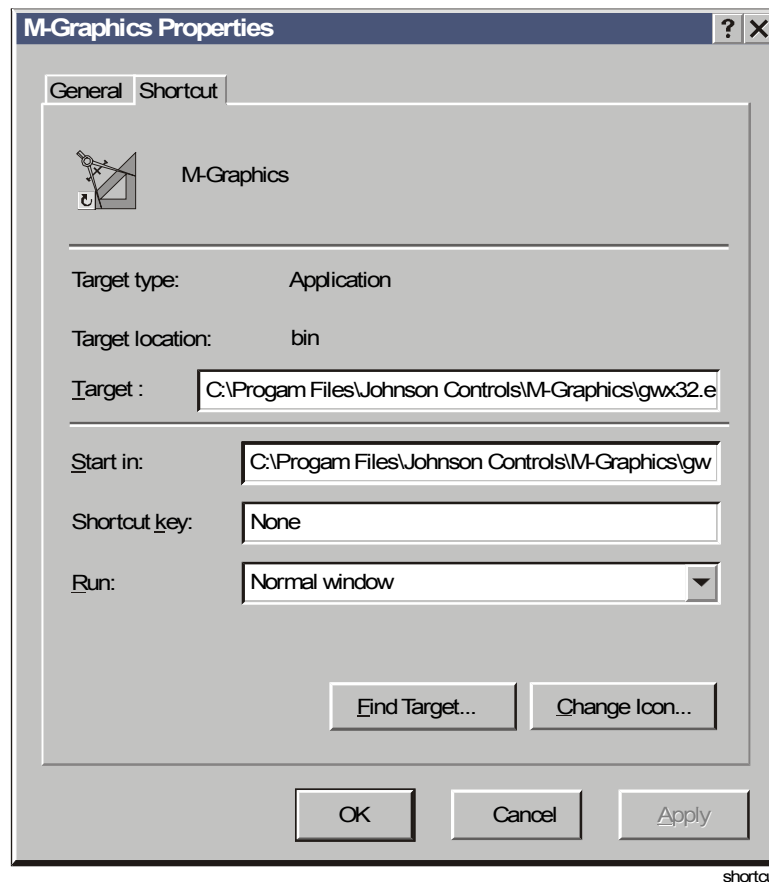


Figure 4-3: Shortcut to M-Graphics Properties Dialog Box

5. Click the Shortcut tab, and enter the appropriate command line options at the end of the Target field. Refer to Table 4-2 and the *Command Line Examples* section.

Table 4-2: Command Line Options

Option	Description
Runtime	Launches M-Graphics in Runtime automatically suppressing Splash screen.
NoSplash	No Splash screen.
RegServer	Registers application.
UnregServer	Unregisters application.

Command Line Examples

In the following command line example, M-Graphics starts with the tanks.gdf display file:

```
C:\Program Files\Johnson Controls\M-Graphics\gwx32.EXE
C:\Program Files\Johnson Controls\M-Graphics\Examples\TANKS.GDF
```

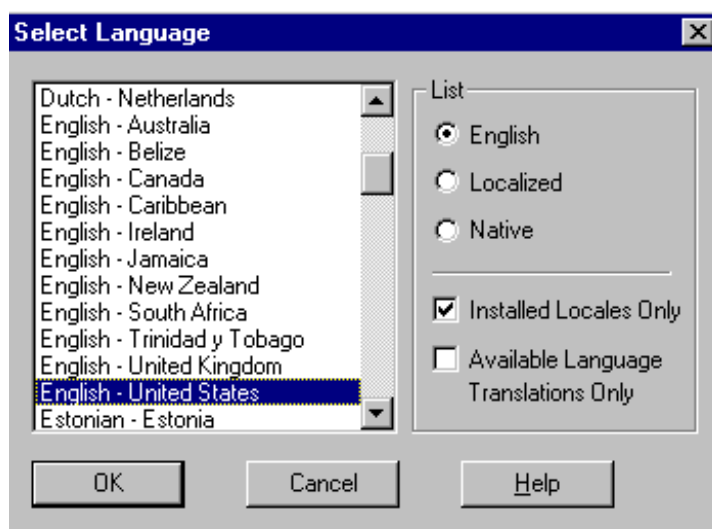
In the following example, M-Graphics starts in Runtime with the tanks.gdf display loaded and the application window in the middle of your screen:

```
C:\Program Files\Johnson Controls\M-Graphics\gwx32.EXE
C:\Program Files\Johnson Controls\M-Graphics\Examples\TANKS.GDF
-Runtime
```

Selecting a Language

To select a language:

1. From the View menu, select Select Language. The Select Language dialog box appears (Figure 4-4).

**Figure 4-4: Select Language Dialog Box**

2. Define the parameters. Refer to Table 4-3.

Table 4-3: Select Language Parameters

Parameter	Description
List	Lists available languages.
Installed Locale Only	Lists all local languages available.
Available Languages Translations Only	List all local languages available to choose.

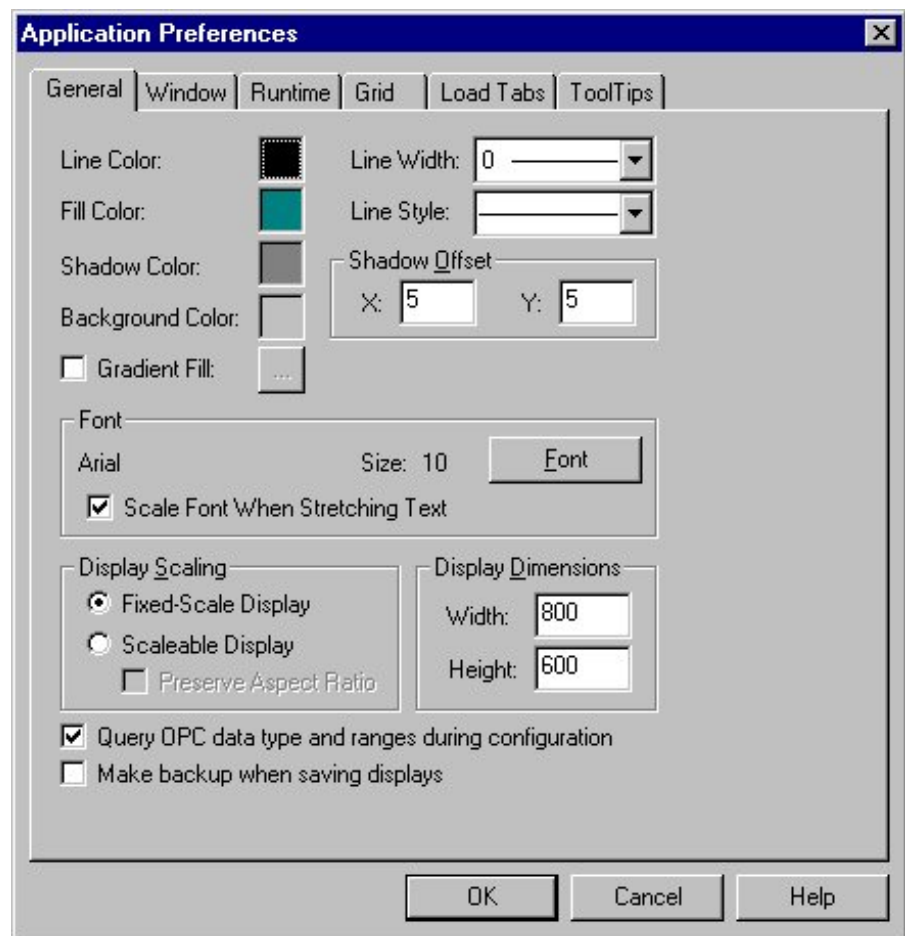
3. Click OK.

Defining General Application Preferences

M-Graphics application preferences apply only to new displays.

To define general application preferences:

1. On the Format menu, select Application Preferences. The Application Preferences dialog box appears.
2. Select the General tab (Figure 4-5).



**Figure 4-5: Application Preferences Dialog Box:
General Tab**

3. Modify the fields and click OK. Refer to Table 4-4.

Table 4-4: General Application Preferences

Field	Description
Line Color	Displays the default line color.
Fill Color	Displays the default fill color.
Shadow Color	Displays the default shadow color.
Background Color	Displays the default background color.
Gradient Fill	Displays the default gradient fill when selected.
Line Width	Displays the default line width.
Line Style	Displays the default line style.
Shadow Offset	Determines the depth of a selected object.
Font	Determines text font and point size.
Display Scaling	Determines whether a display is a Fixed Scale display where the objects do not change size when the main window is resized, or a Scalable Display where objects automatically resize to fit within the main window. The Preserve Aspect Ratio option scales the display proportionate to the screen's aspect ratio.
Display Dimensions	Determines the width and height of the working area (world dimensions) of the display.
Query OPC Data Type and Ranges During Configuration	Searches system for OPC data types and ranges during Configuration of the application.
Make Backup When Saving Displays	Creates a backup file of current display with .bak extension.

If the Query OPC Data Type and Ranges During Configuration box is checked, the system looks for OPC Data Types and Ranges during the configuration of the application.

Defining Window Application Preferences

M-Graphics application preferences apply only to new displays.

To define window application preferences:

1. On the Format menu, select Application Preferences. The Application Preferences dialog box appears.
2. Select the Window tab (Figure 4-6).

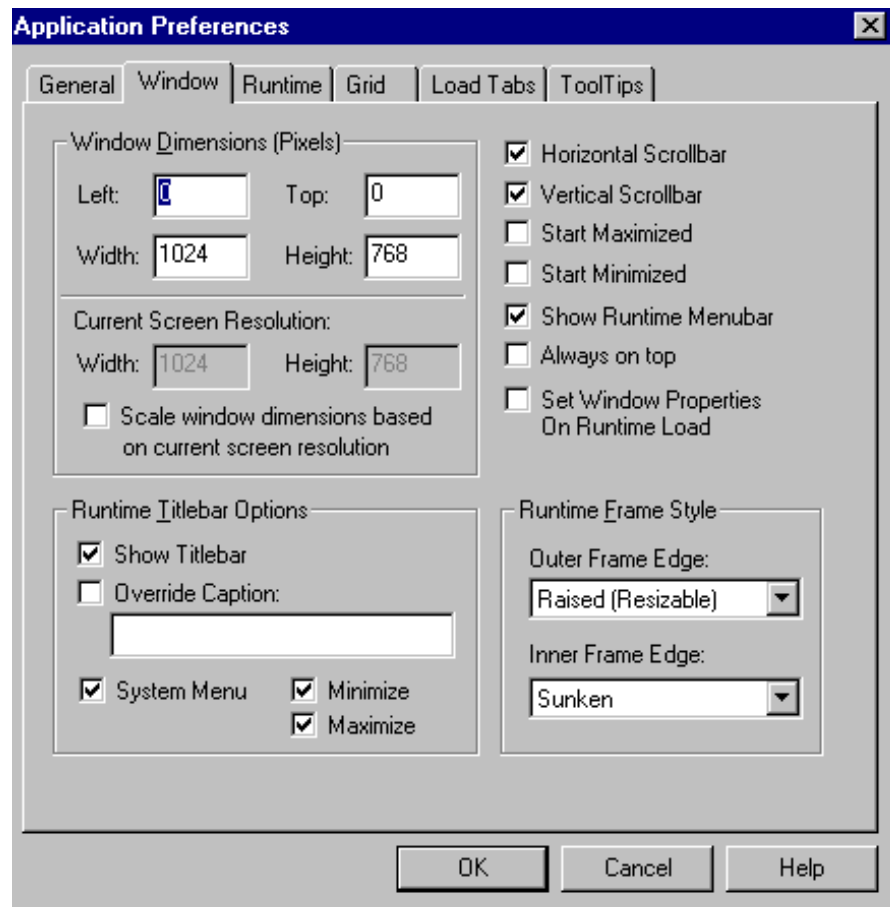


Figure 4-6: Application Preferences: Window Tab

Note: Some of the window settings are only active in Runtime mode.

3. Modify the fields using Table 4-5.
4. Click OK.

Table 4-5: Window Application Preferences

Field	Description
Window Dimensions (Pixels)	Resizes the window to exact dimensions when in Runtime mode.
Current Screen Resolution	<p>Determines screen resolution.</p> <p>If scale window dimensions based on current screen resolution check box is set, the window dimensions are automatically scaled based on the current screen resolution.</p> <p>For example, if the current screen resolution is 640 x 480 and the M-Graphics window size is 640 x 480, and if you switch the resolution to 800 x 600, the next time your display is loaded, its dimensions are scaled up to 800 x 600. If you want the window to always be 640 x 480 no matter what the resolution is, leave this check box unchecked.</p> <p>If the screen resolution is changed, the window dimensions remain the same way they were originally set up.</p>
Runtime Titlebar Options	Hides/shows the title bar by checking the relevant box. Either override the caption, or minimize and maximize the System menu by checking the check boxes.
Horizontal Scrollbar	Hides/shows the horizontal scroll bars in Configure mode and Runtime mode.
Vertical Scrollbar	Hides/shows the vertical scroll bars in Configure mode and Runtime mode.
Start Maximized	Starts with the window initially maximized.
Start Minimized	Starts with the window initially minimized.
Show Runtime Menubar	Hides/shows the Main menu bar in Runtime mode only.
Always on top	Determines if the M-Graphics window is always on top and not obscured by other open windows in Runtime mode only.
Set Window Properties On Runtime Load	<p>Loads window properties for a given display when the display is loaded in Runtime mode.</p> <p>If unchecked, window properties are loaded only when M-Graphics is initially launched into Runtime mode and subsequent display loads retain the window properties of the previously loaded display.</p>
Runtime Frame Style	Specifies styles for the inner and outer edges of the Main window's border.

Defining Runtime Application Preferences

M-Graphics application preferences apply only to new displays.

To define Runtime application preferences:

1. On the Format menu, select Application Preferences. The Application Preferences dialog box appears.
2. Select the Runtime tab (Figure 4-7).

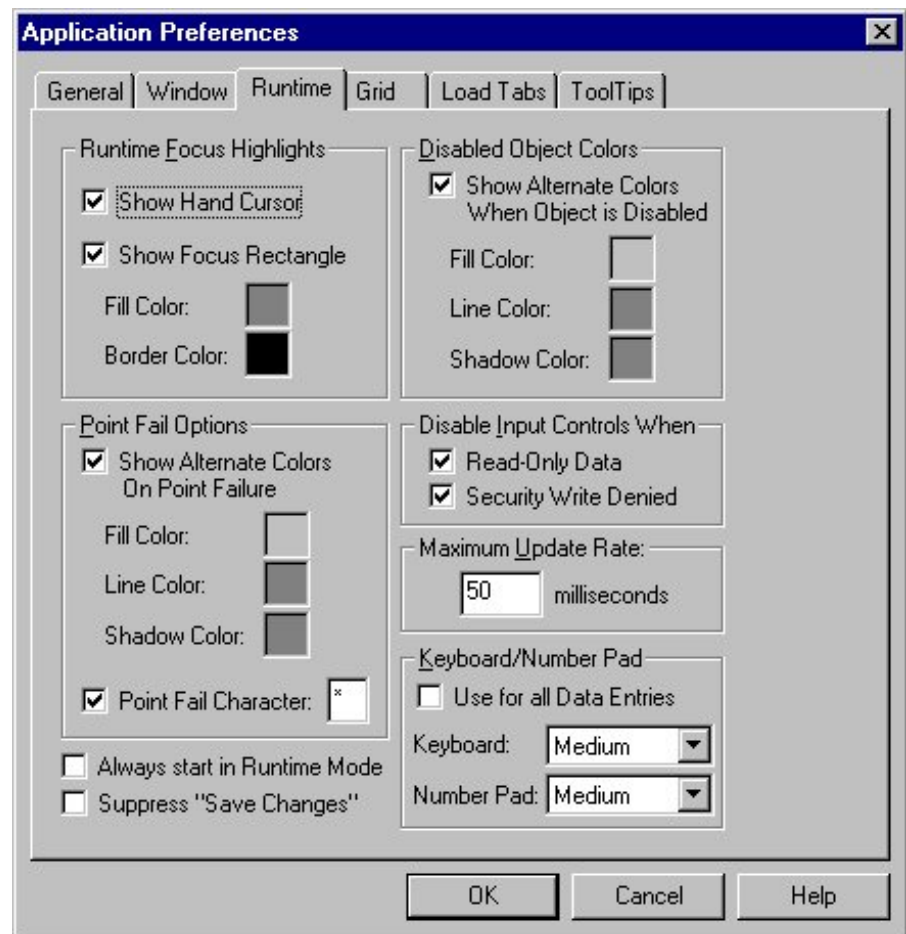


Figure 4-7: Application Preferences: Runtime Tab

3. Modify the fields using Table 4-6.
4. Click OK.

Table 4-6: Runtime Application Preferences

Field	Description
Runtime Focus Highlights	Hides/shows a hand cursor indicating you can pick the object. Hides/shows a frame or focus rectangle around the object able to be picked, and specifies a frame fill color and border color.
Point Fail Options	Determines colors displayed when data points fail. By default, the point fail character shown for process points and data entries is an asterisk.
Disabled Object Colors	Specifies alternate colors shown when the object is disabled. Note you cannot pick disabled objects.
Disable Input Controls When	Disables data input controls automatically when the associated data connection is read-only or when security write access is denied for the associated data connection.
Maximum Update Rate	Specifies how fast Runtime updates can occur with a maximum update rate of 50 milliseconds. Note that OPC server implementations may further restrict the update rate.
Suppress "Save Changes" Message in Runtime	Allows the changes Notification Dialog to be suppressed in Runtime mode.
Start Application in Runtime Mode	Allows application to start in Runtime mode.
Keyboard/Number Pad	Allows the keyboard and Number Pad parameters to be set.

Defining Grid Application Preferences

M-Graphics application preferences apply only to new displays.

To define grid application preferences:

1. On the Format menu, select Application Preferences. The Application Preferences dialog box appears.
2. Select the Grid tab (Figure 4-8).

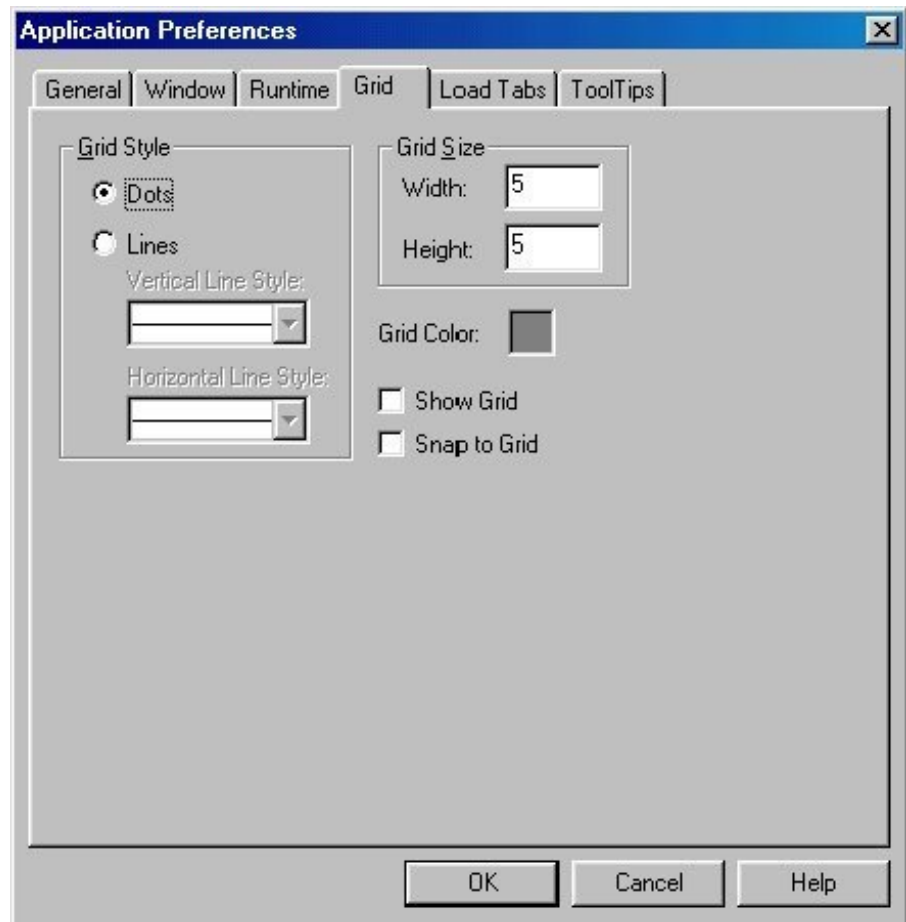


Figure 4-8: Application Preferences: Grid Tab

3. Modify the fields using Table 4-7.
4. Click OK.

Table 4-7: Grid Application Preferences

Field	Description
Grid Style	Specifies if grid lines are dots or lines. If Lines field is selected, determines vertical and horizontal line style.
Grid Size	Specifies the grid spacing in terms of width and height.
Grid Color	Specifies color for the lines/dots of the grid.
Show Grid	Hides/shows grid.
Snap to Grid	Enables/disables snap to grid.

Defining Load Tabs Application Preferences

M-Graphics application preferences apply only to new displays.

To define load tabs application preferences:

1. On the Format menu, select Application Preferences. Application Preferences dialog box appears.
2. Select the Load Tabs tab (Figure 4-9).

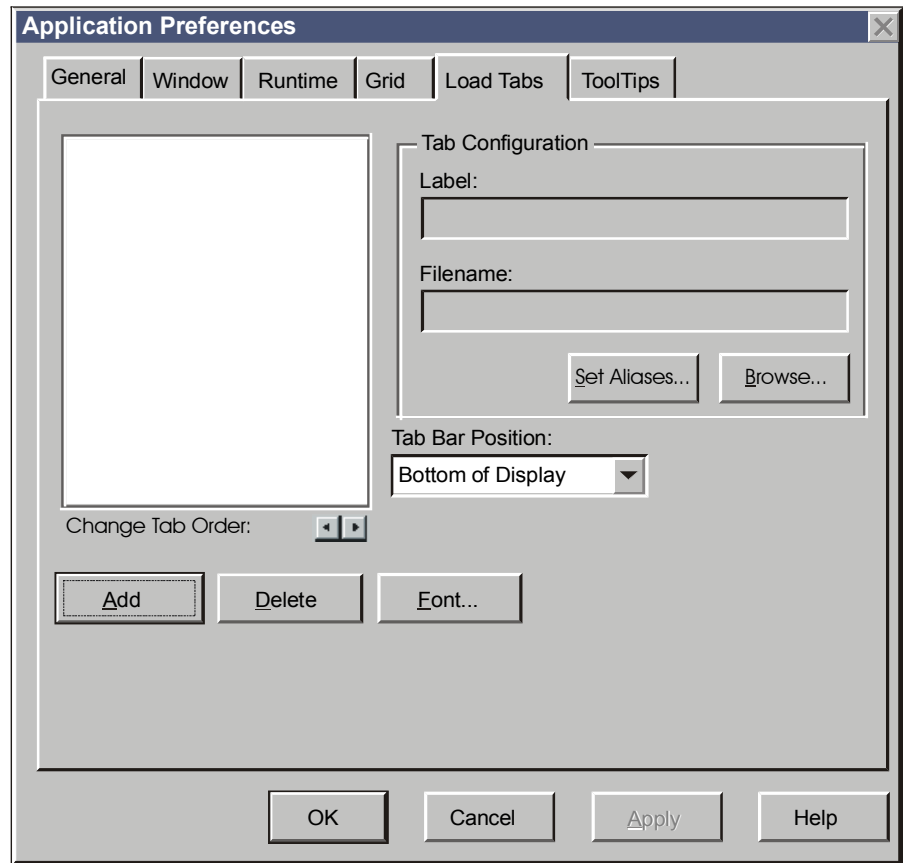


Figure 4-9: Display Properties: Load Tabs Tab

3. Modify the fields using Table 4-8.
4. Click OK.

Table 4-8: Load Tabs Application Preferences

Field	Description
Tab Configuration	Specifies label and filename for tab.
Label	Lists name of tab.
Filename	Determines the display file loaded in tab.
Tab Bar Position	Determines the tab bar position at the top or bottom of the display.
Set Aliases	Sets aliases for the loaded display file.
Browse	Browses displays available to load.
Add	Adds a tab for a file you select. Note you change an existing tab's settings by double-clicking on it.
Delete	Deletes a display tab.
Font	Determines the font for the tab.

Defining ToolTips Application Preferences

M-Graphics application preferences apply only to new displays.

To define ToolTips application preferences:

1. On the Format menu, select Application Preferences. The Application Preferences dialog box appears.
2. Select the ToolTips tab (Figure 4-10).

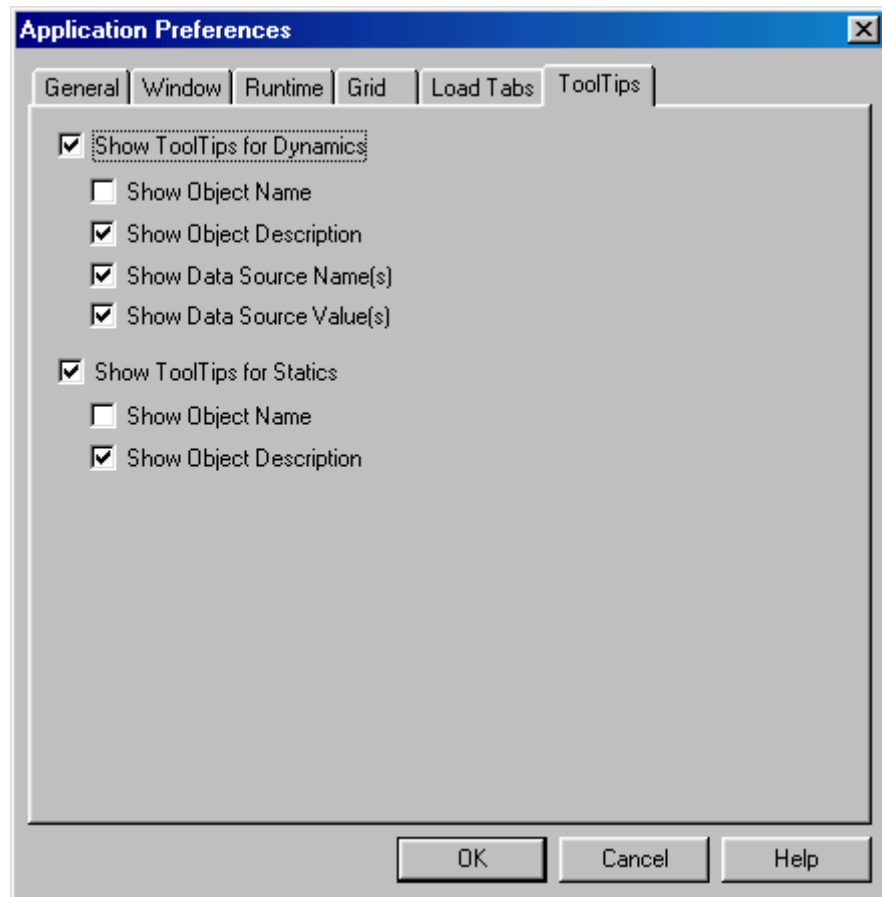


Figure 4-10: Display Properties: ToolTips Tab

3. Modify the fields using Table 4-9.
4. Click OK.

Table 4-9: ToolTips Display Properties

Field	Description
Show ToolTips for Dynamics	Hides/shows ToolTips for dynamics. Options include object name, object description, data source name, data source value.
Show ToolTips for Statics	Hides/shows ToolTips for statics. Options include object name and object description.

Copying Application Preferences to Display Properties

To copy settings from application preferences to display properties:

1. On the Format menu, select Apply Preferences to Properties. The Choose Settings dialog box appears (Figure 4-11).

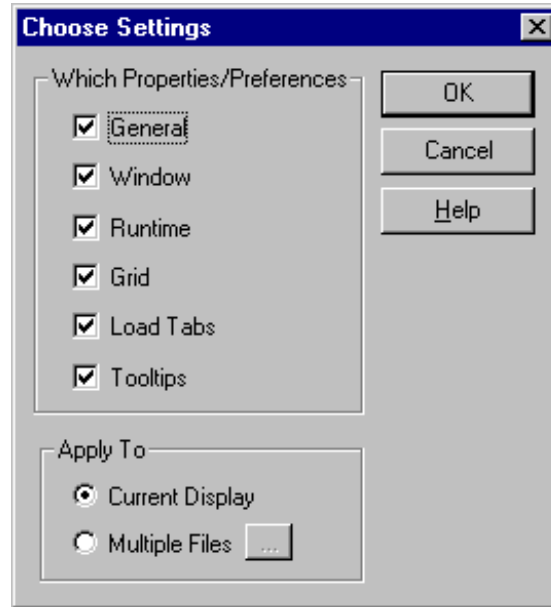


Figure 4-11: Choose Settings Dialog Box

2. Select preference categories.
3. Click OK.

Saving Display Properties as Application Preferences

To save display properties as application preferences:

1. On the Format menu, select Save Properties as Preferences. The Choose Settings dialog box appears (Figure 4-11).
2. Select properties categories.
3. Click OK.

Resetting Default Application Preferences and Display Properties

To reset factory default application preferences and display properties:

On the Format menu, select Reset Default Display Properties or Reset Default Application Preferences.

Defining a Configuration Mode Password

To define a Configuration mode password:

1. On the Format menu, select Configuration Mode Password. The Set Password for Current Display dialog box appears (Figure 4-12).

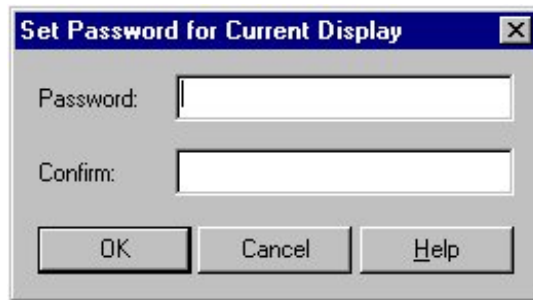


Figure 4-12: Set Password for Current Display

2. Enter the Password.
3. Confirm the Password.
4. Click OK.

Note: When password protected displays are opened, the following Enter Password dialog box appears (Figure 4-13).



Figure 4-13: Enter Password

Setting Up the Alarm Icon Path for ActiveX® Control

To set up the alarm icon path for ActiveX Control:

1. From the Edit menu, select Insert New Object. The Insert Object dialog box appears (Figure 4-14).

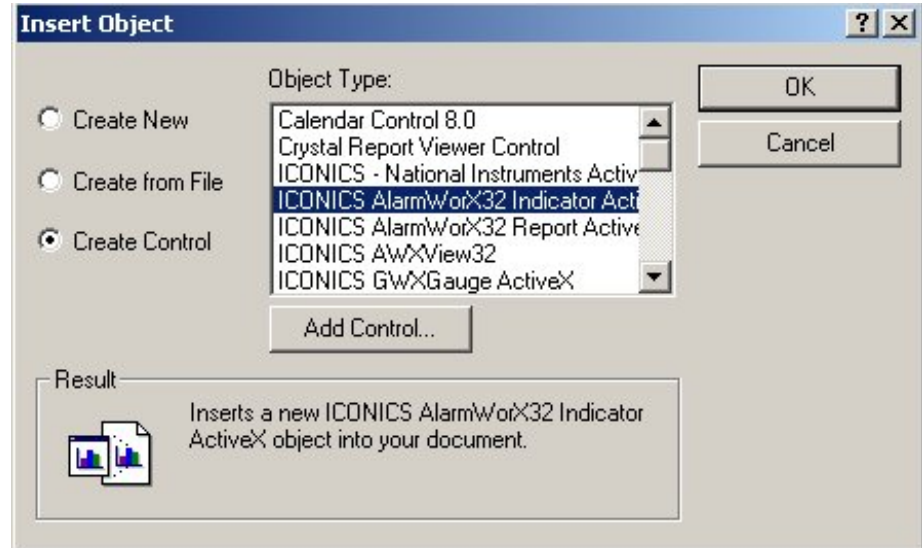


Figure 4-14: Insert Object Dialog Box

2. Select ICONICS AlarmWorX32 Indicator ActiveX.
3. Click OK. The Alarm Icon appears.
4. Double-click on the Alarm Icon. The ActiveX Control Properties dialog box appears.
5. Select the Settings tab (Figure 4-15).

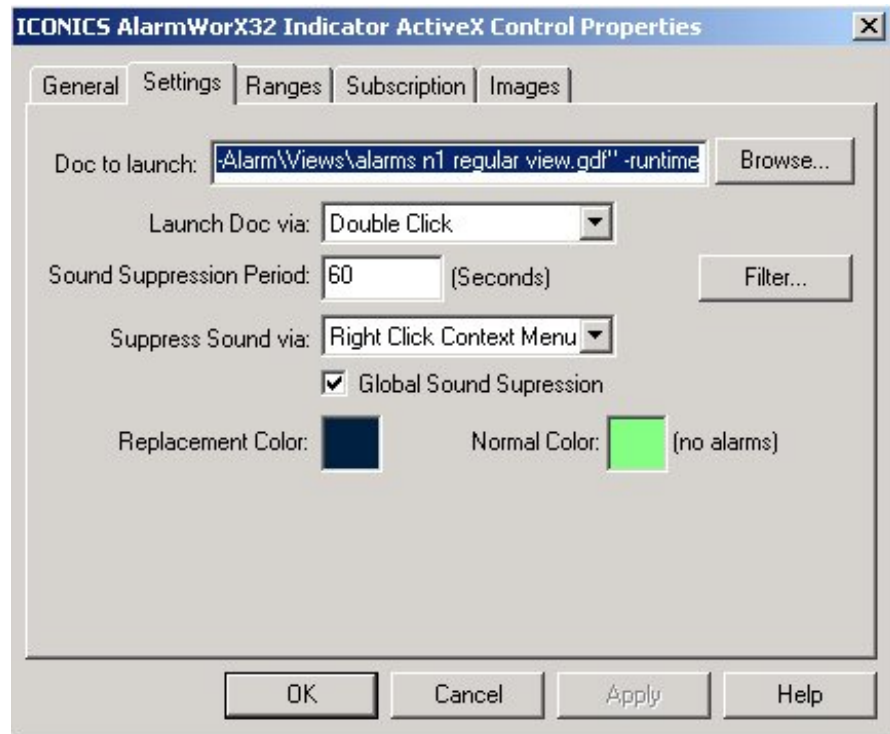


Figure 4-15: ActiveX Control Properties Dialog Box

6. Insert the path to the .exe file in Doc to launch.
7. After the .exe file path, put the actual file name in quotes.
8. Add -runtime at the end of the file name.
9. Click OK.

Chapter 5

Creating Objects

Introduction

This chapter explains how to create and configure objects to use in M-Graphics displays. This chapter describes how to:

- draw a line
- draw a segmented line
- draw a rectangle/square
- draw an ellipse/circle
- insert text
- define text font
- draw an arc
- define line style
- define line width
- fill an object with color
- import bitmaps
- import metafiles
- import symbols
- export a metafile
- freeze an object
- insert an object

Key Concepts

Configure Mode

Configure mode is where you design displays. In this mode, create static and dynamic objects and set general display properties.

Draw Functions

The Draw functions create display objects using various drawing tools. The Draw functions are located in the Draw menu and the Draw toolbar.

M-Graphics Symbols

The main purpose of the Symbol Library is to preview and import previously stored M-Graphics symbols back into M-Graphics. Refer to *Appendix A: OLE Automation Reference*, *Appendix B: DIN Symbol Library*, *Appendix C: French Symbol Library*, and *Appendix D: Johnson Controls Symbol Library* in this manual for examples of symbols included in the Symbol Library. Personal symbols may be saved to the Symbol Library for future use.

Insert Object

This function inserts and embeds an OLE object, such as a chart or an equation, in the current display. The application in which the object was created becomes active on the screen. Insert object also allows you to insert an ActiveX control into the current display. The property sheet for that ActiveX control is automatically opened after inserting the ActiveX control.

Color Palette

The Color Palette defines the color of the object's background, line, or fill (Figure 5-1).



CPalette

Figure 5-1: Color Palette

Color Dialog Box

Double-clicking on the Color Palette displays the Color dialog box (Figure 5-2).

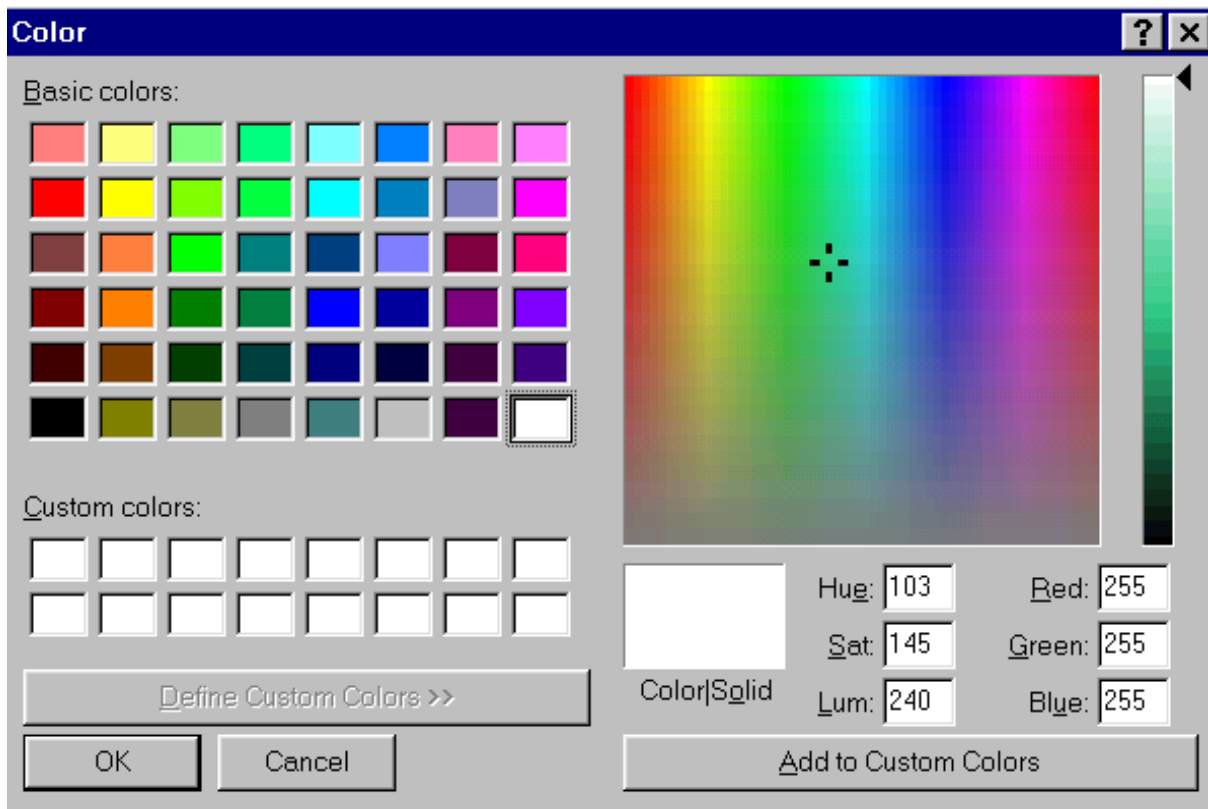


Figure 5-2: Color Dialog Box

Choose from a range of colors and patterns in the color spectrum on the right. Add a custom color or a textured color by clicking the Add to Custom Colors button.

Right-clicking the Color Palette displays a menu, which provides options for the size of the Color Palette dialog box, eyedropper, and resetting default colors.

Wildcards in Find Dialog Box

Wildcard characters '*' and '?' can be used in Find What string. Character '*' can be used to match a group of characters. Character '?' can be used instead of one character. Display all objects from current scope by choosing the Type - Object Name and entering an asterisk (*) to the Find What combo.

Select Handle Color Definitions

When objects or symbols are selected in M-Graphics, colored handles appear around the selection. Table 5-1 describes the color definitions.

Table 5-1: Select Handle Color Definitions

Color of Handle	Description
Blue Handles	The symbol does not have a dynamic tag assigned and is not Frozen.
Red Handles	The symbol has a dynamic assigned.
Green Handles	The symbol is Frozen.
White Handles on All Symbols Except for One	The solid handled is used as a "reference" when using the Arrangement function. The symbol became solid because it was the last symbol in the selected group of symbols.

Text Rotation

Text may be rotated in 90 degree increments by using the Rotate Left and Rotate Right buttons. The rotation ability is available for: multi-line text; left, center, and right alignments; text objects; Process Points (PPTs), data entries, state fields, time/dates, and buttons.

Note: Checkboxes and Radio buttons may not be rotated.

Procedure Overview

Table 5-2: Creating Objects

To Do This	Follow These Steps:
Draw a Line	From the Draw menu, select Straight Line. Left-click on the work area. Drag the mouse and release the button.
Draw a Segmented Line	From the Draw menu, select Segmented Line. Left-click and release the mouse from point to point.
Draw a Rectangle/Square	From the Draw menu, select Rectangle/Square. Left-click and drag to form the box. Pressing Shift while you draw forms a perfectly square box.
Draw an Ellipse/Circle	From the Draw menu, select Ellipse/Circle. Left-click and drag the pointer anywhere in the work area to form an ellipse.
Insert Text	From the Draw menu, select Text. Left-click on the work area. Type the text and left-click outside the text to insert.
Define Text Font	Select the text objects in your display. Select fonts from the Format menu. Define the font parameters and click OK.
Draw an Arc	On the Draw menu, select Arc. Left-click and drag the pointer away from the start point and release.
Define Line Style	Select the object. On the Format menu, select line style. Select a line type.
Define Line Width	Select the object. On the Format menu, select line width. Select a line width.
Fill an Object with Color	Select the object. On the Format menu, select Toggle Fill.
Import Bitmaps	On the Draw menu, select Import > Bitmap. Select a bitmap and click Open.
Import Metafiles	On the Draw menu, select Import > Metafile. Select a metafile and click Open.
Import Symbols	On the Draw menu, select the Import > Symbol. Select a symbol and drag it on the work area. Close the Symbol Library.
Export a Metafile	On the Draw menu, select Export > Metafile. Select a file and click OK.
Freeze an Object	Select the object. On the Format menu, select Toggle Freeze.
Insert an Object	On the Edit menu, select Insert New Object. Choose either Create New, Create from File, or Create Control in the Insert Object dialog box. Select the object type. Click OK.

Detailed Procedures

Drawing a Line

To draw a line:

1. On the Draw menu, select Straight Line. The mouse pointer appears as a pencil.
2. Left-click on the work area.
3. Drag the mouse and release the button.

Note: Hold down the Shift key for a 45° or 90° line.

Drawing a Segmented Line

To draw a segmented line:

1. On the Draw menu, select Segmented Line.
2. Left-click and release the mouse from point to point. Drag the left mouse button for freehand drawing.

Drawing a Rectangle/Square

To draw a rectangle/square:

1. On the Draw menu, select Rectangle/Square. The mouse pointer changes to a box and cross hair.
2. Left-click and drag to form the box.

Note: Pressing Shift while you draw forms a perfectly square box.

Drawing an Ellipse/Circle

To draw an ellipse/circle:

1. On the Draw menu, select Ellipse/Circle.
2. Left-click and drag the pointer anywhere in the work area to form an ellipse.

Note: Pressing Shift while drawing forms a circle.

Inserting Text

To insert text:

1. On the Draw menu, select Text.
2. Left-click on the work area.
3. Type the text and left-click outside the text to insert.

Note: If direct editing, use the Backspace to delete characters. The Delete button deletes all text.

Defining Text Font

To define font types:

1. Select the text objects in your display.
2. Select fonts from the Format menu. The Font dialog box appears.
3. Define the font parameters and click OK. The text appears with the font parameters you defined.

Drawing an Arc

To draw an arc:

1. On the Draw menu, select Arc.
2. Left-click and drag the pointer away from the start point and release.

Defining Line Style

To define line style:

1. Select the object (line, rectangle, segmented line, ellipse, or arc).
2. On the Format menu, select line style.
3. Select a line type.

Note: Line styles only exist for line widths of 0.

Defining Line Width

To define the line width:

1. Select the object (line, rectangle, segmented line, ellipse, or arc).
2. On the Format menu, select line width.
3. Select a line width.

Filling an Object with Color

To fill an object with color:

1. Select the object.
2. On the Format menu, select Toggle Fill.

Note: Lines, bitmaps, and metafile images cannot be filled or unfilled with color.

Importing Bitmaps

To import bitmaps:

1. On the Draw menu, select Import > Bitmap (.bmp).
2. Select a bitmap and click Open.

Importing Metafiles

To import metafiles:

1. On the Draw menu, select Import > Metafile (.emf, .wmf).
2. Select a metafile and click Open.

Importing Symbols

To import symbols:

1. On the Draw menu, select the Import > Symbol.
2. Select a symbol and drag it on the work area.
3. Close the Symbol Library.

Exporting a Metafile

To export a metafile:

1. On the Draw menu, select Export > Metafile (.emf).
2. Select a file and click OK.

Freezing an Object

To freeze an object:

1. Select the object.
2. On the Format menu, select Toggle Freeze. A lock appears on the cursor when over the object.

Inserting an Object

To insert an object:

1. On the Edit menu, select Insert New Object. The Insert Object dialog box appears (Figure 5-3).

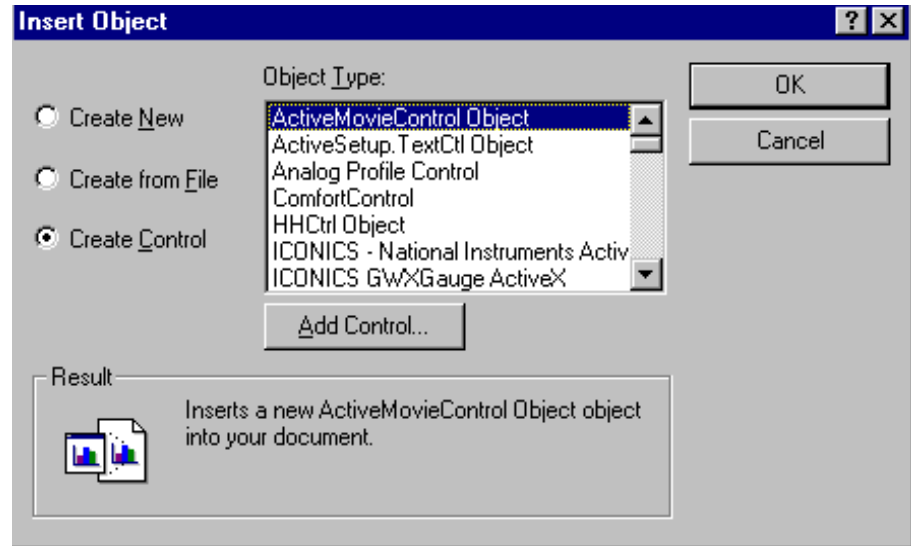


Figure 5-3: Insert Object Dialog Box

2. Choose either Create New, Create from File, or Create Control in the Insert Object dialog box.
3. Select the object type.
The Add Control button allows you to add a control to the existing list. To create control, specify a control from the list of controls displayed at the right.
4. Click OK.

Troubleshooting

Common Problems

Table 5-3: Troubleshooting Objects

Situation	Solution
For M-Graphic displays or symbols built using impacted Metafile images, memory leaks occur causing depletion of computer system resources. This results in degraded performance and potential system lock up.	<p>Do not use embedded Metafile images or rebuild/convert the Metafile images. (See note.)</p> <p>To rebuild:</p> <ol style="list-style-type: none"> 1. Right-click on the graphic image (or symbol). 2. Under the Popup menu, select Convert To Symbol to convert the graphic image to an M-Graphics symbol (format). 3. Save the changes. 4. Run the M-Graphics display.
An error appears evaluating expressions in a pop-up balloon when the cursor is over a symbol with an action. This problem is caused by the N1OPC Server intermittently losing its connection to the network.	Reboot the M5 Workstation PC to restore the network connection.
Text typed in the default font/size becomes smaller in Runtime mode, and remains smaller after returning to Configure mode.	<p>To make the font/size of text the same in Runtime and Configure modes:</p> <ol style="list-style-type: none"> 1. Select all objects. Reselect the Arial font and type 10 in the size field. 2. Deselect the Stretch Text On Resize property in the Property Inspector – Text tab. 3. Deselect the Scale Font When Stretching Text property in the Display Properties – General tab. 4. Select the Scaleable Display and Preserve Aspect Ratio properties in the Display Properties – General tab.
<p>Note: This problem is not unique to M-Graphics but is a general Metafile format issue based on how the Metafile images are originally created.</p>	

Chapter 6

Editing Objects

Introduction

This chapter explains how to edit objects in M-Graphic displays. This chapter describes how to:

- edit the length of a line
- reposition a line
- edit the vertices of a segmented line
- resize a rectangle
- resize an ellipse
- resize an arc
- edit the shape of an arc
- cut an object
- copy an object
- paste an object
- use Paste Special
- duplicate an object
- find an object
- replace an object
- report an object

Key Concepts

Edit Functions

The Edit functions create display objects using various editing tools. While editing graphics, stay at 100% Zoom to maintain a reference for all graphic files. This helps minimize the possibility of creating a graphic that is not sized properly when in the Runtime mode. The Edit functions are located in the Edit menu.

Paste vs. Paste Special

When pasting objects into M-Graphics, the Paste Special function is recommended. Paste Special is used for pasting objects in a specified format when more than one format is available. The ability to paste an object as a bitmap or Metafile or device independent bitmap is available in the Paste Special dialog box.

The Paste function places a copy of the objects currently in the clipboard into the work area in native form. The object remains on the clipboard until it is copied or cut. This command is used to insert a copy of the clipboard contents at the insertion point.

Find vs. Replace Functions

The Find function operates the result in the display as a selection, and it shows the result as object hierarchy in the tree view information. Both functions display the same dialog box options.

Note: The Replace function replaces the Find What strings with Replace With string.

Procedure Overview

Table 6-1: Editing Objects

To Do This	Follow These Steps:
Edit the Length of a Line	Select the line. Press the left mouse button on a square handle of the line. Drag the mouse to extend the line and release the mouse button.
Reposition a Line	Select the line. Drag to a new position and release the mouse button.
Edit the Vertices of a Segmented Line	Select the line. Press Shift and right-click. Edit the midpoints and vertices of the segments.
Resize a Rectangle	Select the rectangle, left-click on a square handle, and drag to resize. Use the Shift key to maintain the Aspect Ratio of the rectangle while resizing.
Resize an Ellipse	Select the ellipse. Left-click on a square handle and drag it to resize the ellipse.
Resize an Arc	Select the arc. Press the left mouse button on a square handle, drag it to reshape the arc, and release.
Edit the Shape of an Arc	Right-click on the arc to display the Format menu and choose Edit Arc.
Cut an Object	Select the object to cut. On the Edit menu, select Cut.
Copy an Object	Select an object. On the Edit menu, select Copy.
Paste an Object	On the Edit menu, select Paste.
Use Paste Special	On the Edit menu, select Paste Special. Select the format for the object, paste, or paste link. Click OK.
Duplicate an Object	Select an object. On the Edit menu, select Duplicate.
Find an Object	On the Edit menu, select Find. Fill in the fields. Click OK.
Replace an Object	On the Edit menu, select Replace. Fill in the fields. Click OK.
Report an Object	On the Edit menu, select Report. Fill in the fields. Click OK.

Detailed Procedures

Editing the Length of a Line

To edit the length of a line:

1. Select the line.
2. Press the left mouse button on a handle of the line.
3. Drag the mouse to extend the line and release the mouse button. Hold down the Shift key for a 45° or 90° line.

Repositioning a Line

To reposition a line:

1. Select the line.
2. Drag to a new position and release the mouse button.

Editing the Vertices of a Segmented Line

To edit the vertices of a segmented line:

1. Select the line.
2. Press Shift and right-click. The line's vertices appear, and the mouse cursor appears as a cross arrow.
3. Edit the midpoints and vertices of the segments.

Resizing a Rectangle

To resize a rectangle:

1. Select the rectangle, left-click on a square handle, and drag to resize.
2. Use the Shift key to maintain the Aspect Ratio of the rectangle while resizing.

Note: Rounded rectangles may be free-rotated and have 3D edges.

Resizing an Ellipse

To resize an ellipse:

1. Select the ellipse.
2. Left-click on a square handle and drag it to resize the ellipse.

Note: Pressing Shift while resizing maintains the Aspect Ratio of the object.

Resizing an Arc

To resize an arc:

1. Select the arc.
2. Press the left mouse button on a square handle, drag it to reshape the arc, and release.

Editing the Shape of an Arc

To edit the shape of an arc:

Right-click on the arc to display the Format menu and choose Edit Arc. The mouse pointer changes to a cross hair and allows you to change the shape of the arc.

Cutting an Object

To cut an object:

1. Select the object to cut. A box with square handles surrounds the object.
2. On the Edit menu, select Cut. The object is removed from the display and sent to the clipboard.

Note: You can also use the Delete key to cut selected objects and place them onto the clipboard.

Copying an Object

To copy an object:

1. Select an object. A box with handles surrounds the object.
2. On the Edit menu, select Copy. The object is copied to the clipboard.

Pasting an Object

To paste an object from the clipboard:

On the Edit menu, select Paste. The object is pasted near the center of the M-Graphics display.

Using Paste Special

To use Paste Special:

1. On the Edit menu, select Paste Special. The Paste Special dialog box appears (Figure 6-1).

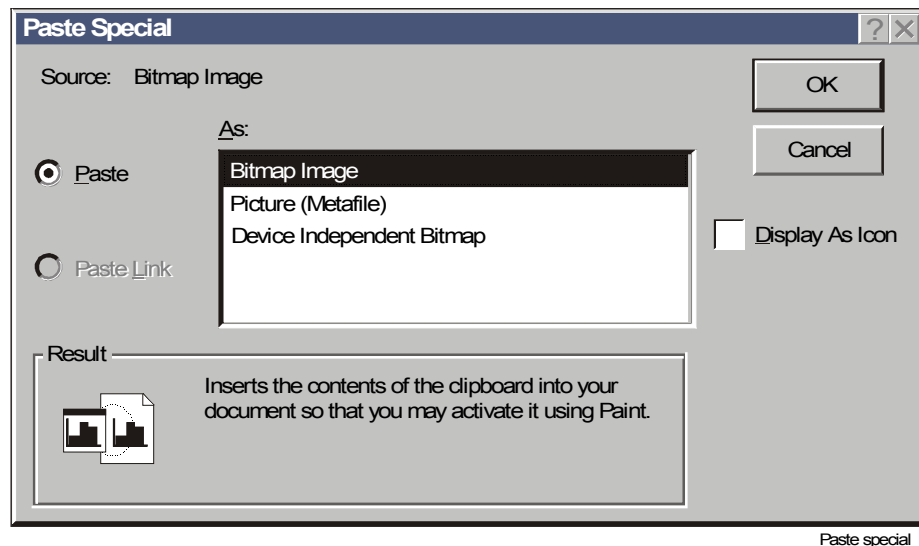


Figure 6-1: Paste Special Dialog Box

2. Select the format for the object, paste, or paste link, and decide if you want to display the object as an icon.
3. Click OK.

Duplicating an Object

To duplicate an object:

1. Select an object. A box with square handles surrounds the object.
2. On the Edit menu, select Duplicate.

Finding an Object

To find an object:

1. On the Edit menu, select Find. The Find dialog box appears (Figure 6-2).

The Advance window displays the tree with the results of the Find inquiry. Click Apply instead OK to see that information in the Advance Window.

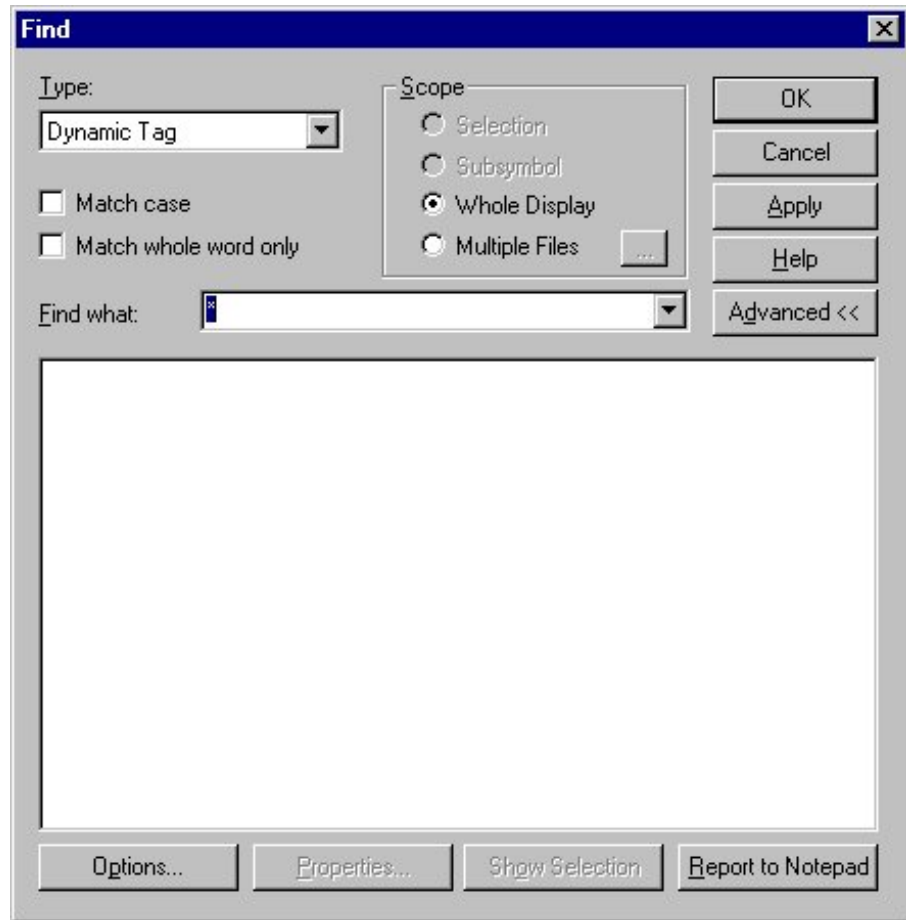


Figure 6-2: Find Dialog Box

2. Fill in the fields using Table 6-2.
3. Click OK.

Note: The Find dialog box is the same as the Report dialog box.

Table 6-2: Find/Replace/Report Dialog Box Fields

Parameter	Description
Type	Allows you to specify a type of string where the Find/Replace works. All the strings can be modified for every object/dynamic action.
Dynamic Tag	Allows you to find/replace tags (dynamics points).
Text Label	Allows you to find/replace labels of text objects.
Object Name	Allows you to find/replace names of objects/dynamic actions.
Keyword	Allows you to find/replace global keywords used in shared objects.
Scope	Determines the operating space for the operation.
Selection	Works in current selection; if there are no objects selected, option is disabled.
Subsymbol	Works in subsymbol (see menu item Edit Symbol and Subsymbol Editing capability); if there is no subsymbol edited, this option is disabled.
Whole Display	Works in whole display; any subsymbol editing is left and replaced by a root edit level.
Multiple Files	Works in a group of file displays; any subsymbol editing is left and replaced by root level. If there are changes made, you are prompted to save the document.
Match case	If selected, the function finds/replaces text strings that match the case of the characters. Otherwise the command finds/replaces strings with either uppercase or lowercase.
Match whole word only	If checked, the function finds/replaces the whole text strings. Otherwise Find/Replace selects any string.
Find what	Specifies the search text. A drop-down menu lists possible strings. Wildcards are allowed. Note if the type is Object Name and the Find What string is '*', the tree view displays all object hierarchy from current scope including multiple files scope.
Replace with	Specifies the string to replace the characters found. A drop-down menu lists possible strings. This set of strings is determined by the Scope and by the Type radios. Wildcards do not work here.
Tags Menu	Opens the Expression Editor, Local Variables, or Simulation Variables dialog box.
Advanced	Toggles between a big dialog box when the tree view and the Show button are visible and a small dialog box.
Options...	Opens Report Options dialog box and provides the following options for inclusion in report tree: object name, keyword, dimensions, user description, custom data, dynamic actions, and dynamic tags. The Report Options dialog box determines how the items are enclosed: double quote, quote, or nothing.
Show Sel	Highlights the item selection found in the list.
Properties	Displays Property dialog box for highlighted item.
Report to Notepad	Displays results of query in Notepad.

Replacing an Object

To replace an object:

1. On the Edit menu, select Replace. The Replace dialog box appears (Figure 6-3).

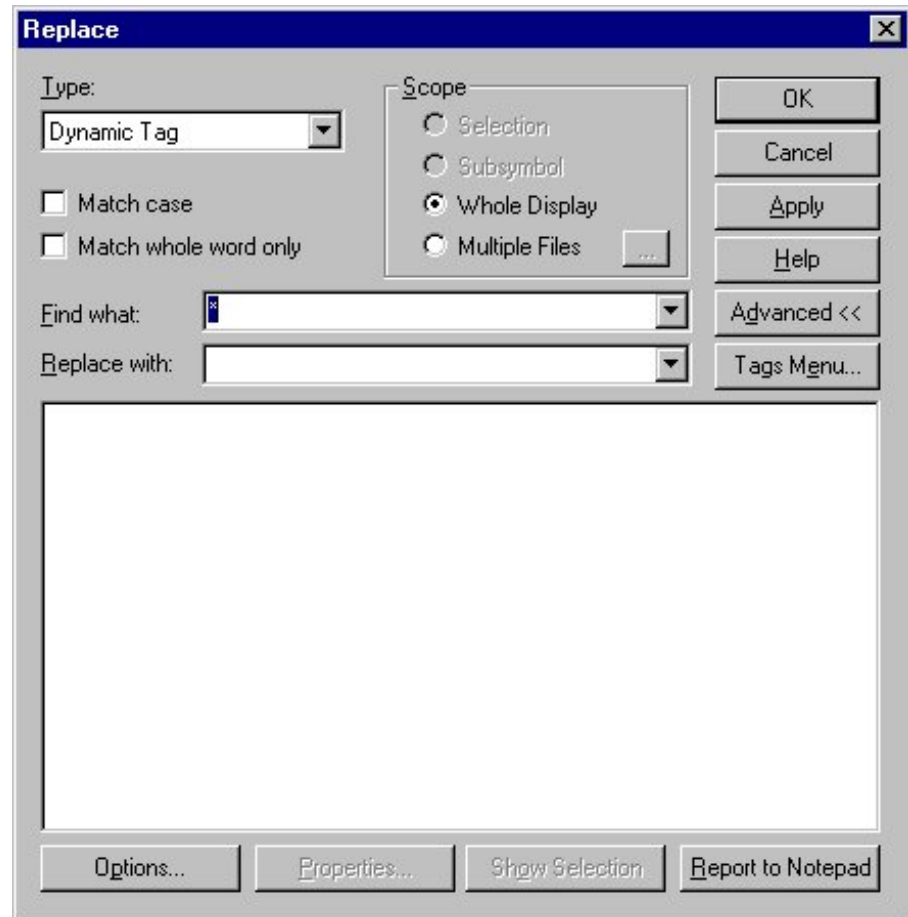


Figure 6-3: Replace Dialog Box

2. Fill in the fields using Table 6-2.
3. Click OK.

Reporting an Object

To report an object:

1. On the Edit menu, select Report. The Report dialog box appears (Figure 6-4).

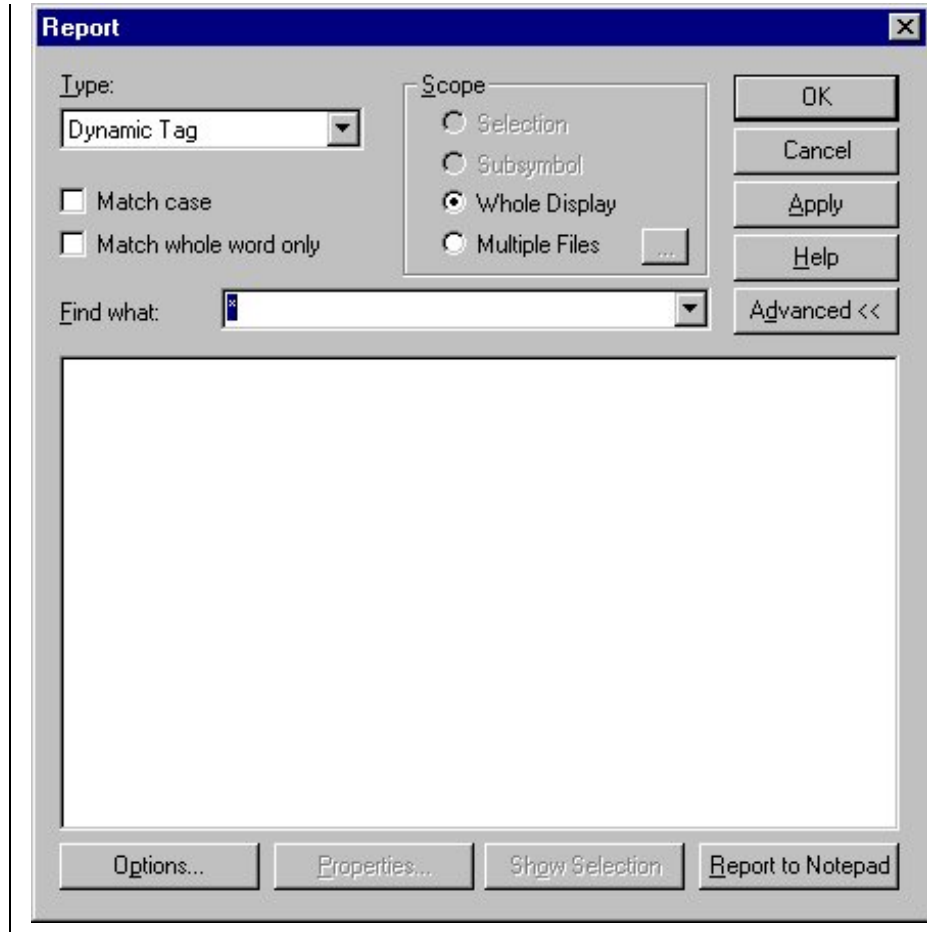


Figure 6-4: Report Dialog Box

2. Fill in the fields using Table 6-2.
3. Click OK.

Chapter 7

Arranging Objects

Introduction

M-Graphics provides several features to arrange objects in the display. This chapter describes how to:

- group objects
- ungroup objects
- reposition objects
- rotate objects
- flip objects
- align objects
- space objects evenly
- make objects the same size
- add layers
- remove layers
- duplicate layers
- edit layer properties
- set currently active layer

Key Concepts

Layers

The use of layers in M-Graphics is helpful since layers allow the elements in the display to be categorized, separating the display into levels of detail through decluttering. It is possible to add, remove, or duplicate a layer as well as edit the layer properties, set the active layer, set the next layer, set the previous layer, hide layers above the current layer, and hide layers below the current layer.

Layer Configuration

Every M-Graphics display starts as a display with one layer, the primary or system layer. This first layer is the only layer to which a template can be applied. When dealing with a display that has several layers, the attribute picked for the primary layer appears in all layer views. While the objects placed in the system layer can be seen in the configuration of all other layers, it is not possible to edit these components while editing other layers. Layer names indicate what the layer contains and why it is a separate layer.

Figure 7-1 is an example of the Edit Layer Properties dialog box.

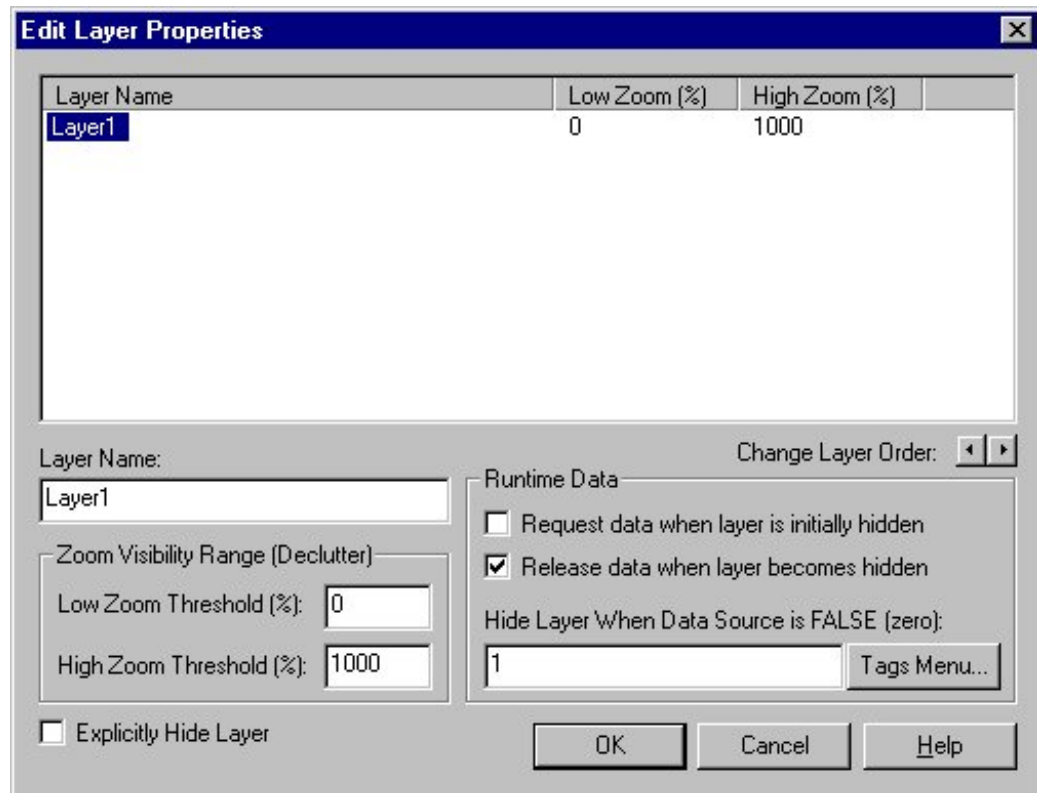


Figure 7-1: Edit Layers Properties Dialog Box

Zoom Visibility Range

By assigning each layer a different zoom threshold, decluttering occurs. Decluttering is automatically showing/hiding elements depending on the current zoom status. Zoom thresholds cannot be applied to the primary/system layer. The default values for zoom threshold are 0-1000, which makes the layer always visible. By changing this threshold for specific layers, the zoomed out view only shows the most basic display.

Procedure Overview

Table 7-1: Arranging Objects

To Do This	Follow These Steps:
Group Objects	Select objects. On the Arrange menu, select Group into Symbol.
Ungroup Objects	Select a group. On the Arrange menu, select Ungroup Symbol.
Reposition Objects	Select an object. On the Arrange menu, select one of the following: bring to front, send to back, bring forward, or send backward.
Rotate Objects	Select an object. On the Arrange menu, select Free Rotate and click and drag on a handle or Rotate/Flip and one of the following: rotate left or rotate right.
Flip Objects	Select an object. On the Arrange menu, select Rotate/Flip and choose either flip horizontal or flip vertical.
Align Objects	Left-click on the objects. On the Arrange menu, select Align and choose one of the following: tops, bottoms, middles, lefts, rights, or centers.
Space Objects Evenly	Left-click on the objects. On the Arrange menu, select Space Evenly and choose either across or down.
Make Objects Same Size	Left-click on the objects you wish to make a uniform size. On the Arrange menu, select Make Same Size and choose one of the following: height, width, or both.
Add Layers	On the Format menu, select Layers > Add Layer. Refer to Table 7-2.
Remove Layers	On the Format menu, select Layer > Remove Layer. Select the layer to remove. Click OK.
Duplicate Layers	On the Format menu, select Layer > Duplicate Layer. Select the layer to duplicate. Click OK.
Edit Layer Properties	On the Format menu, select Layers > Edit Layer Properties. Select desired layer. Edit layer properties using Table 7-2. Click OK when finished.
Set Currently Active Layer	On the Format menu, select Layers > Set Currently Active Layer. Select desired layer. Click OK.

Detailed Procedures

Grouping Objects

To group objects:

1. Select objects.
2. On the Arrange menu, select Group into Symbol.

Note: Individual objects are unable to be edited while grouped.

Ungrouping Objects

To ungroup objects:

1. Select a group.
2. On the Arrange menu, select Ungroup Symbol.

Repositioning Objects

To reposition an object:

1. Select an object.
2. On the Arrange menu, select one of the following: bring to front, send to back, bring forward, or send backward.

Rotating Objects

To rotate objects:

1. Select an object.
2. On the Arrange menu, select Free Rotate and click and drag on a handle or Rotate/Flip and one of the following: rotate left or rotate right.

Note: Text, bitmaps, and metafiles cannot be rotated. Objects that have been free-rotated can also be flipped.

Flipping Objects

To flip objects:

1. Select an object.
2. On the Arrange menu, select Rotate/Flip and choose either flip horizontal or flip vertical.

Aligning Objects

To align objects:

1. Left-click on the objects.
2. On the Arrange menu, select Align and choose one of the following: tops, bottoms, middles, lefts, rights, or centers.

Spacing Objects Evenly

To space objects evenly:

1. Left-click on the objects.
2. On the Arrange menu, select Space Evenly and choose either across or down.

Making Objects the Same Size

To make objects the same size:

1. Left-click on the objects you wish to make a uniform size.
2. On the Arrange menu, select Make Same Size and choose one of the following: height, width, or both.

Adding Layers

To add a layer:

1. On the Format menu, select layers > Add Layer. The Edit Layer dialog box appears (Figure 7-2).

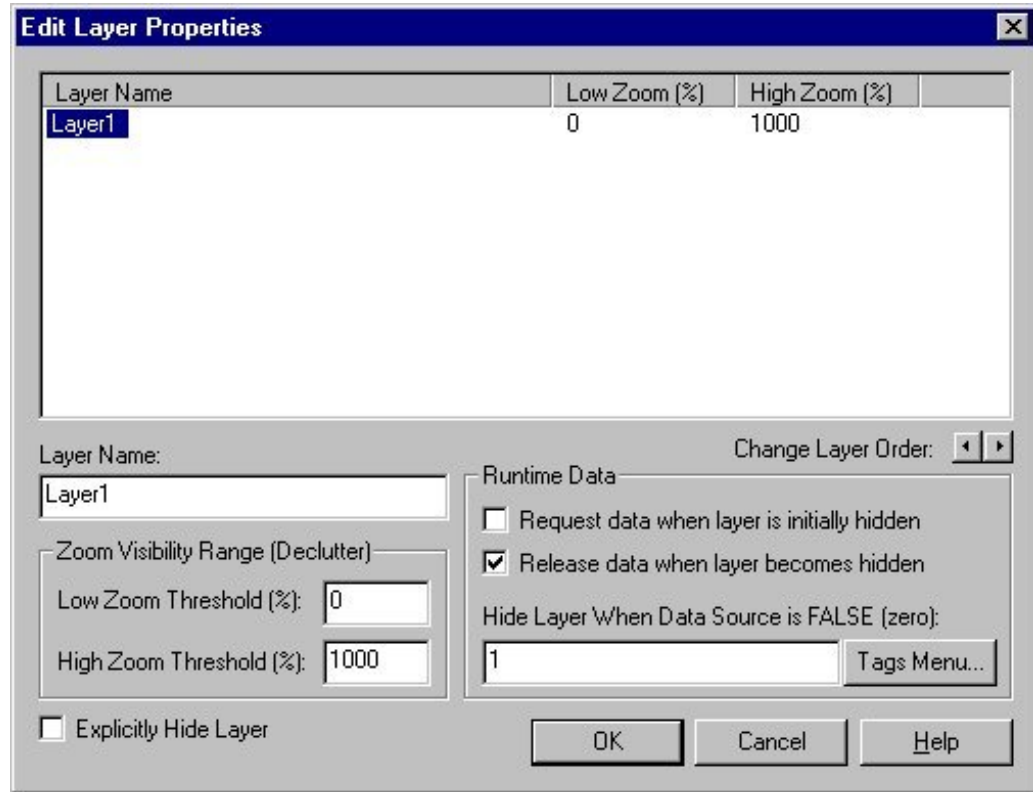


Figure 7-2: Edit Layer Properties Dialog Box

2. Refer to Table 7-2.

Table 7-2: Edit Layer Properties Parameters

Field	Description
Layer Name	Enter layer name indicating what the layer contains or why the layer is separate.
Change Layer Order	Move selected layer up or down in the list.
Zoom Visibility Range (Declutter)	Assign Zoom Threshold to declutter the display. Elements of each layer automatically hide or show depending on the current zoom percent. The default Zoom Threshold is 0-1000.
Runtime Data	Set how data should be retrieved from the OPC Server during Runtime. To retrieve data when the layer is hidden, Check <i>Request data when layer is initially hidden</i> . To release data when layer becomes hidden, check <i>Release data when layer becomes hidden</i> .
Explicitly Hide Layer	When checked, forces the layer to remain hidden even if the zoom threshold indicates the layer is visible. Layer remains hidden until this option is turned off (unchecked). Note: If a check appears to the left of the layer name, that layer is hidden during runtime.

Removing Layers

To remove a layer:

1. On the Format menu, select Layer > Remove Layer. The Remove Layer dialog box appears (Figure 7-3).

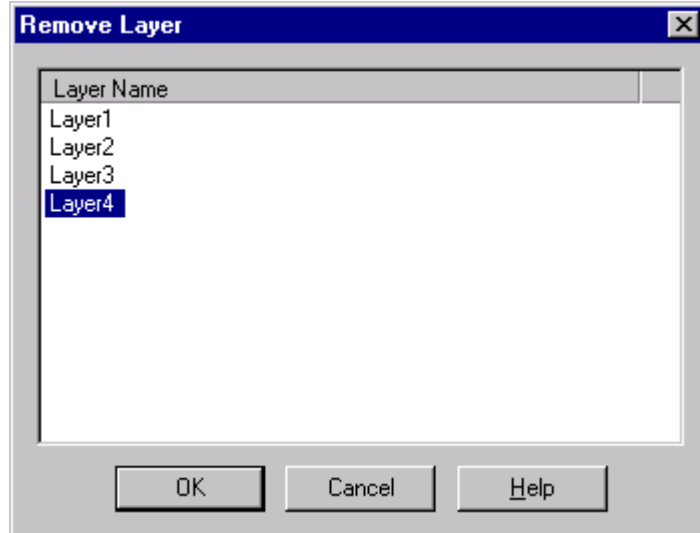


Figure 7-3: Remove Layer Dialog Box

2. Select the layer to remove.
3. Click OK.

Duplicating Layers

To duplicate a layer:

1. On the Format menu, select Layer > Duplicate Layer. The Duplicate Layer dialog box appears (Figure 7-4).

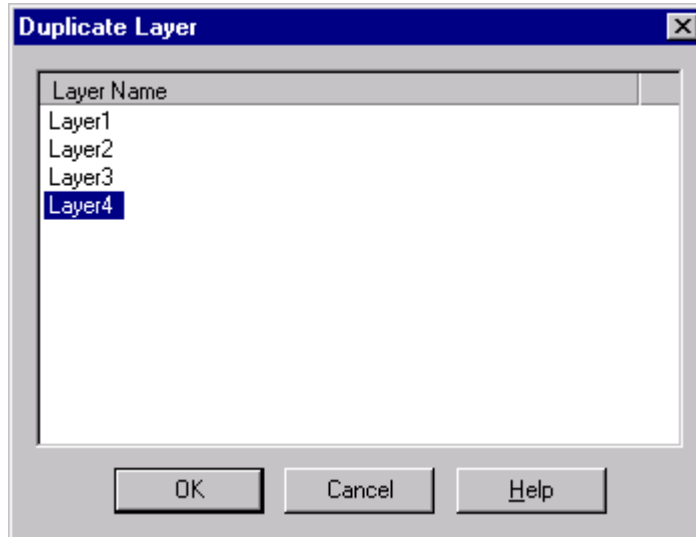


Figure 7-4: Duplicate Layer Dialog Box

2. Select the layer to duplicate.
3. Click OK.

Editing Layer Properties

To edit layer properties:

1. On the Format menu, select Layers > Edit Layer Properties. The Edit Layer Properties dialog box appears (Figure 7-5).

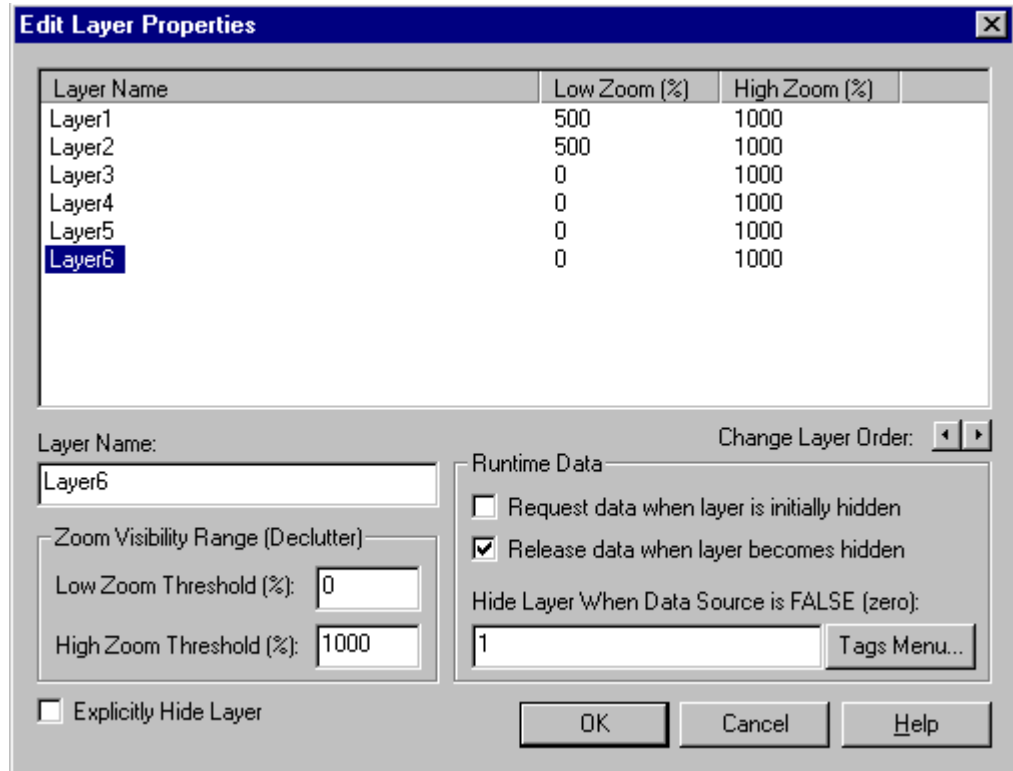


Figure 7-5: Edit Layers Properties Dialog Box

2. Select desired layer.
3. Edit layer properties using Table 7-2.
4. Click OK when finished.

Setting Currently Active Layer

To set currently active layer:

1. On the Format menu, select Layers > Set Currently Active Layer. The Set Currently Active Layer dialog box appears (Figure 7-6).

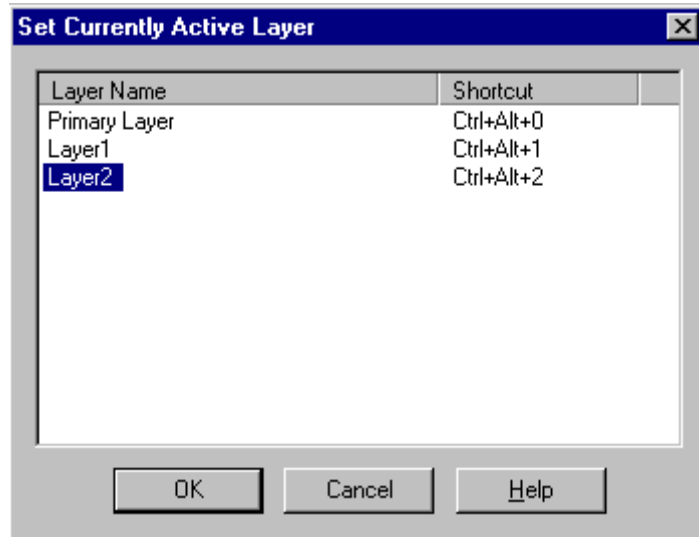


Figure 7-6: Set Currently Active Layer Dialog Box

2. Select desired layer.
3. Click OK.

Note: Table 7-3 describes options available after setting the current active layer.

Table 7-3: Setting Currently Active Layer Options

Option	Description
Set Active Layer Next	Sets the active layer after the currently selected layer.
Set Active Layer Previous	Sets the active layer before the currently selected layer.
Hide Layers Above Current Layer	Hides all layers above the currently active layer.
Hide Layers Below Current Layer	Hides all layers below the currently active layer.

Chapter 8

Using Templates

Introduction

Templates allow users to create a framework that saves time when creating new display files. This chapter describes how to:

- create a template
- insert template objects
- apply a template
- remove an applied template
- edit an applied template
- update template displays

Key Concepts

Templates

Templates are created by saving a display with a template file extension (.tdf). Templates can contain anything a standard display contains including static, dynamics objects, standard color, and shapes. Templates save time by allowing you to create a framework that can be applied to all your display files. Templates are essentially backgrounds useful for implementing a consistent and standard look in graphic files.

Update Template Displays

The update template displays function is used to globally update displays that have the same template file applied to them as the template file currently being updated. Changes made while editing a .tdf file will be applied to displays based on that .tdf file.

Template Object

Template objects are essentially placeholder frames, which are later replaced by a desired object. There are currently three types of template objects: bitmaps, metafiles, and OLE objects. When a template object is added to a display, it shows a text message like `Double-click Here to Add Bitmap`. When you double-click on the object, you are prompted for a bitmap file to replace the template object at its current size and location. This feature can be used to lay out generic displays, which can be specialized later by filling in the placeholders for bitmaps, metafiles, and OLE objects. The Set Original Dimensions feature restores a bitmap to its actual dimensions after it has been stretched to another size.

Procedure Overview

Table 8-1: Using Templates

To Do This	Follow These Steps:
Create a Template	On the File menu, select New. Modify file as desired. On the File menu, select Save. Enter file name and choose template display .tdf as the file type.
Insert Template Objects	On the Format menu, select Template > Insert Template Object. Select bitmap, metafile, or OLE object. Click OK.
Apply a Template	On the Format menu, select Template > Apply Template. Select a template file (.tdf extension) and click Open. Any current Template is replaced. The template is placed behind any object currently on the screen.
Remove an Applied Template	On the Format menu, select Template > Remove Applied Template.
Edit an Applied Template	On the Format menu, select Template > Edit Applied Template. After modifications to the template are finished, on the Format menu, select Template > Cancel Edit Applied Template to exit Edit Applied Template mode and return to Normal mode.
Update Template Displays	On the Format menu, select Template > Update Template Displays. Enter the files for updating. Click OK.

Detailed Procedures

Creating a Template

To create a template:

1. On the File menu, select New.
2. Modify file as desired.
3. On the File menu, select Save. The Save dialog box appears.
4. Enter file name and choose template display .tdf as the file type.

Inserting Template Objects

To insert template objects:

1. On the Format menu, select Template > Insert Template Object. The Choose Template Object Type dialog box appears.
2. Select bitmap, metafile, or OLE object.
3. Click OK. A placeholder for the selected object type is placed in the template.

Applying a Template

To apply a template:

1. On the Format menu, select Template > Apply Template. The Open dialog box appears.
2. Select a template file (.tdf extension) and click Open. Any current Template is replaced. The template is placed behind any object currently on the screen.

Note: Open the Summary window to determine if a template is attached.

Removing an Applied Template

To remove an applied template:

On the Format menu, select Template > Remove Applied Template.

Editing an Applied Template

To edit an applied template:

1. On the Format menu, select Template > Edit Applied Template.

Note: Only template objects are visible in Edit Applied Template mode.

2. After modifications to the template are finished, on the Format menu, select Template > Cancel Edit Applied Template to exit Edit Applied Template mode and return to Normal mode.

Updating Template Displays

This function is only available when editing a .tdf file.

To update template displays:

1. On the Format menu, select Template > Update Template Displays.
2. Enter the files for updating.
3. Click OK.

Note: Make sure the Working Directory points to the folder that contains the *.tdf file in order to update the *.gdf files.

Chapter 9

Using the Property Inspector

Introduction

This chapter describes the Property Inspector. This chapter describes how to:

- edit object parameters in the Property Inspector
- update shared objects

Key Concepts

Property Inspector

The Property Inspector dialog box lets you view and change object parameters.

The Property Inspector contains several tabs. The primary tab is General, which reflects the properties of a given static object. The Property Inspector may also display additional tabs, which show information about attached dynamics. (In Figure 9-1, the text object has a size action, a flash action, and a hide action.)

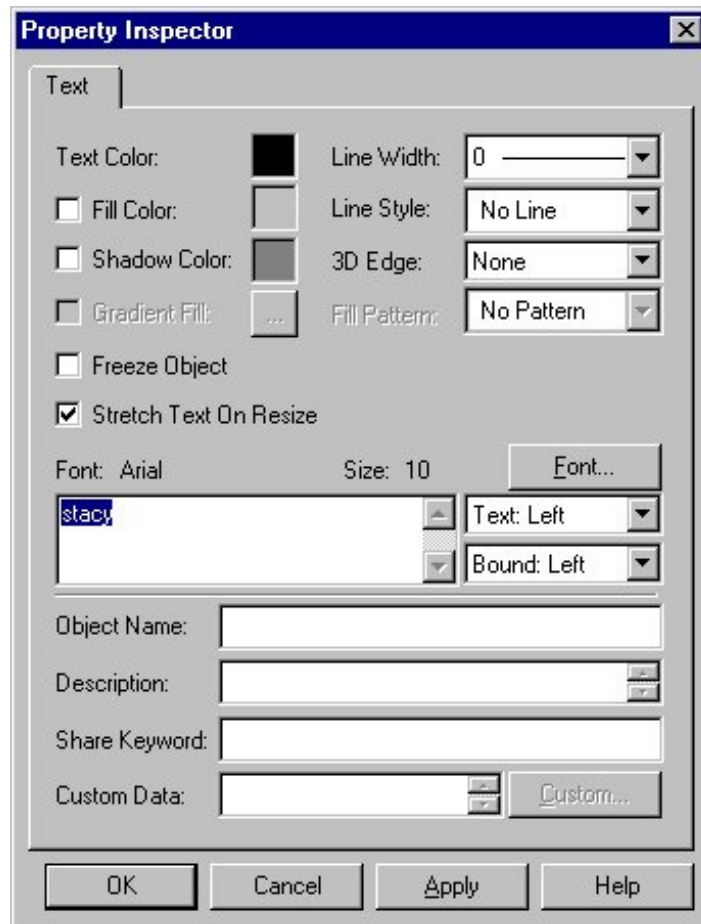


Figure 9-1: Property Inspector Dialog Box: Text Tab

Update Shared Objects

Enter a keyword shared by all the objects in the Share Keyword field. Any change can be made to an object of the group and can be applied to the rest of objects by using Update Shared Objects (Figure 9-2).

Tips on Using Shared Objects

Consider a display with many inputs and outputs. If all input label objects are marked as shared, e.g., with input keyword, and all output label objects are marked as shared e.g., with output keyword, it is very fast and easy to modify the color, size, 3D effect, and shadow for all of them. Change one object and apply the change to the rest.

Consider a project with many pumps. If all desired pumps are marked as shared with same keyword, one can quickly and easily change the appearance. If the pumps are connected to different signals, make sure Preserve Dynamic Tags is checked. For changes on tags, consider using Find/Replace with wildcards to the desired group of objects to be searched/replaced.

Consider a company logo or similar pictures placed in many displays in different positions. By one operation, all of the displays can be modified. For example, to increase in size, make one logo bigger, select Update Function, and don't forget to uncheck Preserve Object's Size.

Procedure Overview

Table 9-1: Using the Property Inspector

To Do This	Follow These Steps:
Edit Object Parameters in the Property Inspector	Double-click on an object. Edit the parameters and click OK to return to the work area.
Update Shared Objects	On the Edit menu, select Update Shared Objects. Fill in the fields.

Detailed Procedures

Editing Object Parameters in the Property Inspector

- To edit object parameters in the Property Inspector:
- 1. Double-click on an object. The Property Inspector dialog box appears (Figure 9-1).
 - 2. Edit the parameters and click OK to return to the work area.
- The fields of the Property Inspector dialog box may vary slightly depending on what type of object is being inspected.

Updating Shared Objects

- To update shared objects:
- 1. On the Edit menu, select Update Shared Objects. The Update Shared Objects dialog box appears (Figure 9-2).

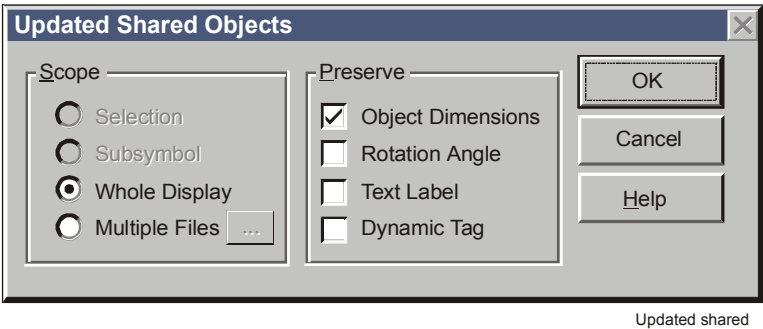


Figure 9-2: Update Shared Objects Dialog Box

- 2. Fill in the fields using Table 9-2.

Table 9-2: User Interface for Update Shared Objects

Parameter	Description
Scope	Determines the operating space for the operation.
Selection	Works in current selection; if there are no objects selected, this option is disabled.
Subsymbol	Works in subsymbol; if there is no subsymbol edited, this option is disabled.
Whole Display	Works in whole display; any subsymbol editing is left and replaced by a root edit level.
Multiple Files	Works in a group of file displays; any subsymbol editing is left and replaced by root level. If there are changes in the current document, a dialog appears to save the changes or cancel the operation.
Preserve	Preserve (not update) any of the following attributes by checking the appropriate check boxes: Object Dimensions, Rotation Angle, Text Label, and Dynamic Tag.

Chapter 10

Establishing Data Source Connections

Introduction

Dynamic animation is achieved by transforming visible objects based on specified data source connections. This chapter describes how to:

- connect to a data source
- operate the OPC Universal Tag Browser
- add a simulated variable
- edit expressions
- edit local variables
- create an Alias
- edit an Alias
- set Runtime Alias

Key Concepts

Data Source Connections

Each dynamic object allows one or more data source connections for its primary values. Some dynamics also allow range overrides; these range overrides are also data source connections.

A data source connection is:

- a Tag representing a value in the system
- a constant value
- a local M-Graphics variable
- an Alias
- a mathematical expression (which can include Tags, constants, and local variables)
- a simulated equation

At Runtime, users can drag/drop source support and browse local variables and Aliases. Type Data Source connections directly into the Data Source field in the Property Inspector dialog box of the various dynamics (Figure 10-1) or click the OPC Tags to open the OPC Universal Tag Browser (Figure 10-2).

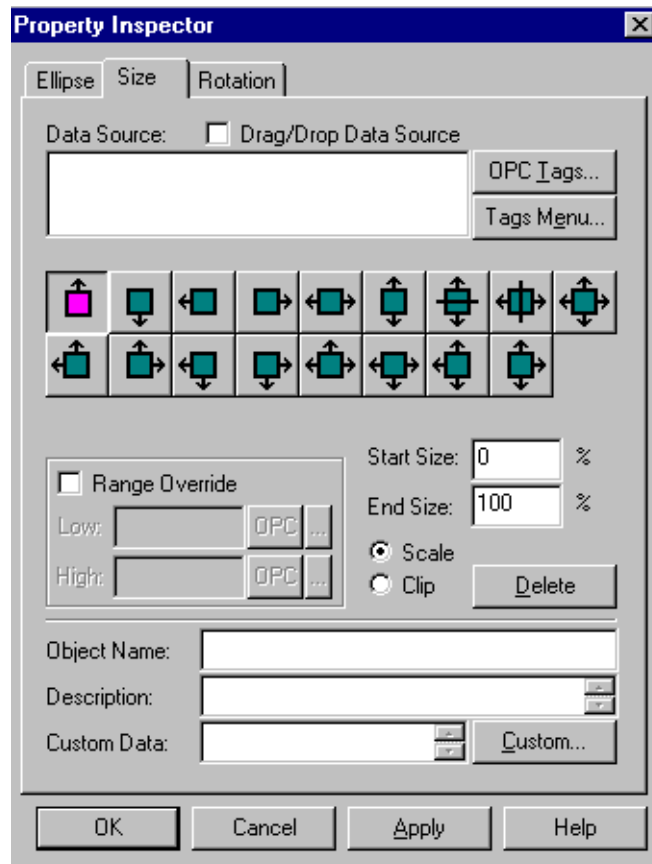


Figure 10-1: Property Inspector: Size Dynamic

OPC Universal Tag Browser

Clicking the OPC Tags button in the Property Inspector opens the OPC Universal Tag Browser. The OPC Universal Tag Browser allows you to configure OPC data points and lists current connected Tags.

Tree View

The tree view presents a hierarchical display of information. The view includes information on network location, server name, and group hierarchy. The view allows you to create a point name based on server location and group names. This allows points from the OPC data servers to be configured from a remote client PC. Outside of network connections, a special setup is not required to support remote browsing.

OPC Universal Tag Browser Dialog Box

Figure 10-2 is an example of the OPC Universal Tag Browser.

Table 10-1 describes the areas of the OPC Universal Tag Browser.

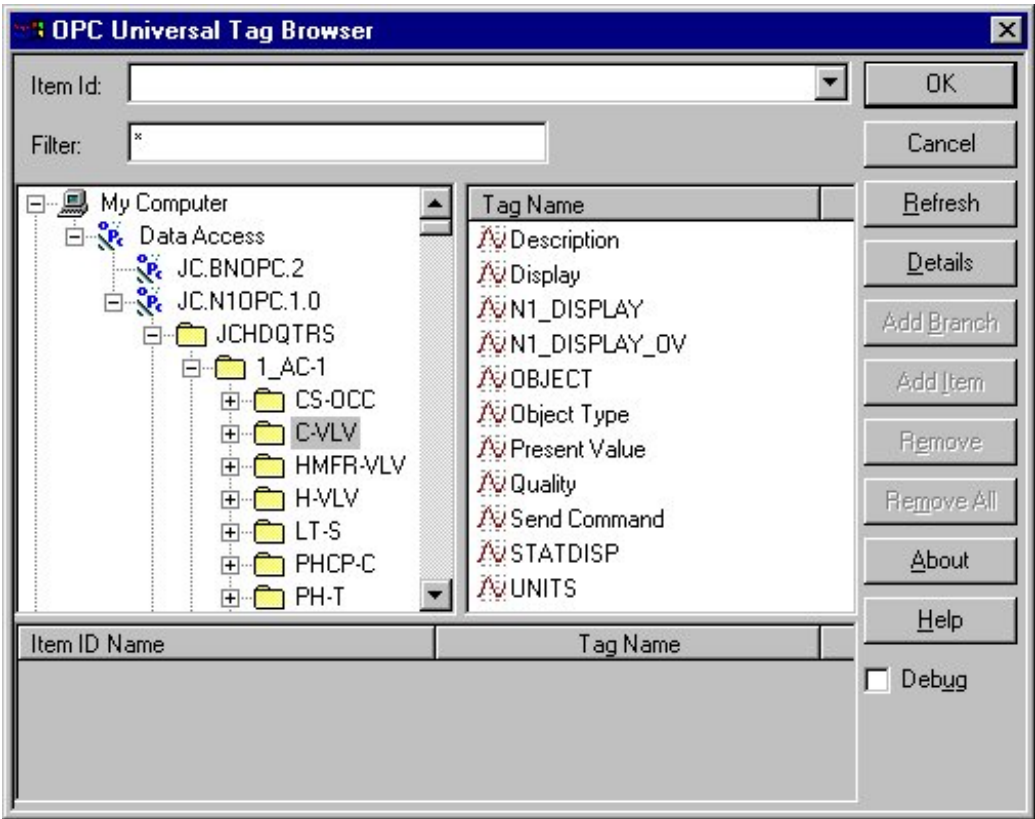


Figure 10-2: OPC Universal Tag Browser Dialog Box

Table 10-1: Areas of the OPC Universal Tag Browser

Areas of the OPC Universal Tag Browser	Description
Item ID	Shows the full Tag name. A list of previously selected points is available in the drop-down list.
Filter	Sends a filter to the OPC server.
Tag Name	Displays the full Tag name in Item ID box if you double-click or highlight Tag Names list.
Refresh	Displays updated network information.
Details	Displays information about specific item.
Add Branch	Allows an additional branch to be added to the tree diagram.
Add Item	Allows an additional item to be added.
Remove	Allows selected item to be removed.
Remove All	Allows everything in the Tag Name to be deleted.
Debug	Debugs selected Tag when checked.

Tags

All non-numeric data connection strings, which do not include the special tokens as described later in this section, are interpreted as Tags. Any Tags that are not defined in the system at Runtime fail to connect i.e., {{JC.CFOPC\1.ACM B17\required.PresentValue}}.

If a Tag happens to be a string that is a number (a rare situation) and you don't want the Tag to be auto-detected as a constant value, use the following syntax:

```
{{Tag_name}}
```

Constant Values

M-Graphics automatically recognizes if an entered data connection string is a number and interprets that data connection as a constant value. If you want to enter a constant string value (a non-number constant), use the following syntax:

```
$"constant_string_value"$ i.e., $"1.75"$
```

This syntax prevents the string from being auto-detected as a Tag name.

Local M-Graphics Variables

To indicate that a data connection is a local variable, use the following syntax:

~~local_variable_name~~

The scope of local M-Graphics variables is confined to the display in which they are defined. Typically, local variables would be used as values for animation effects (i.e., simulated values that do not need to be defined at the system level).

State Fields

State fields refer to the state of a variable. Normally, state fields are binary values each attached to a string having some type of meaning.

Figure 10-3 is an example of the State Field Configuration dialog box.

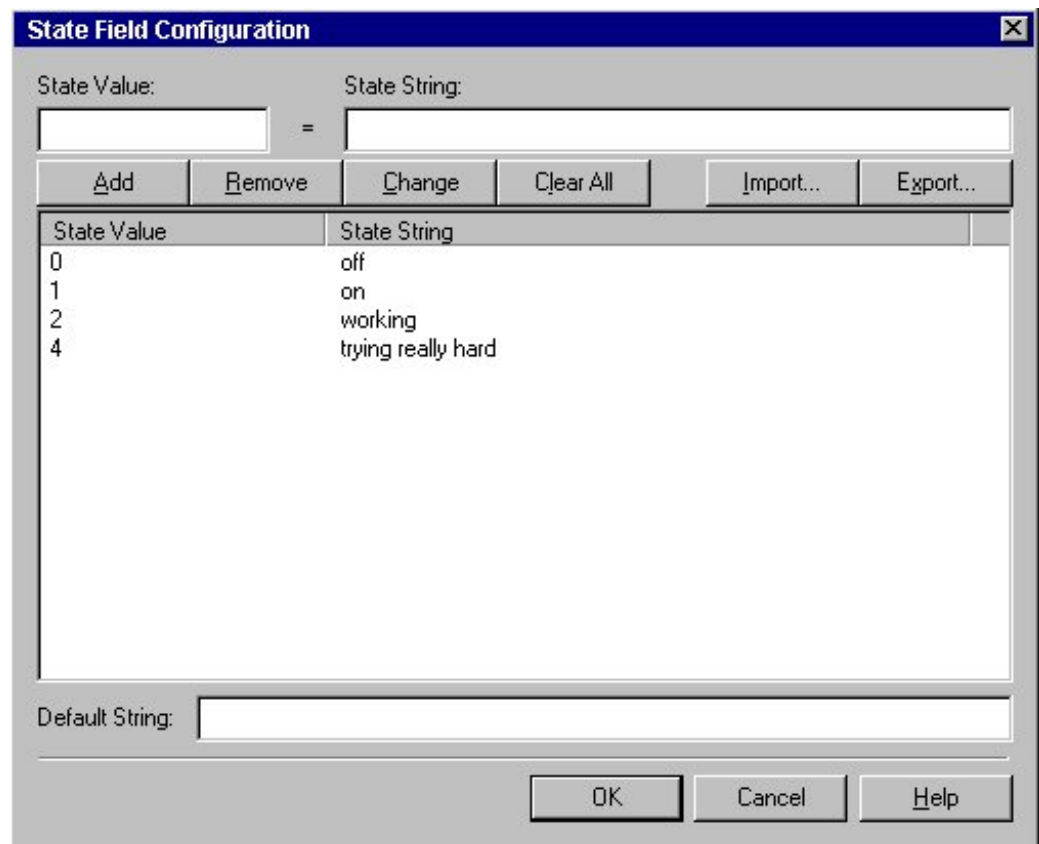


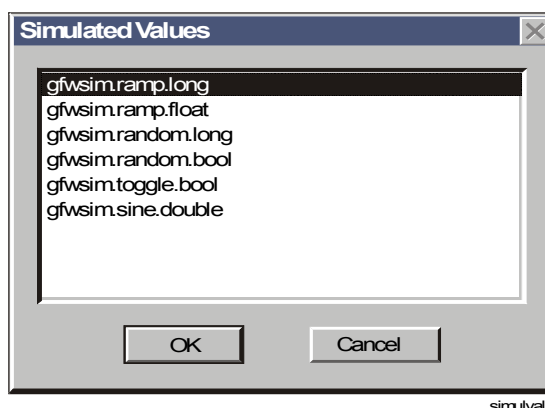
Figure 10-3: State Field Configuration Dialog Box

Table 10-2: State Field Configuration Attributes

Field	Description
State Value	Displays the current value.
State String	Displays the current string associated with the value.
Add	Adds state value that appears in State Value box.
Remove	Removes selected state value.
Change	Changes selected state value.
Clear All	Clears all state values on list.
Import	Displays the Open or Save As dialog box.
Export	Displays the Open or Save As dialog box.
Default String	Displays the string that appears when state does not have a pre-configured string.

Simulated Values

M-Graphics includes several predefined local simulation variables. These simulation variables are useful for testing display animations when an OPC server is unavailable (Figure 10-4).



simulval

Figure 10-4: M-Graphics Simulated Values Dialog Box

Expressions

Expressions allow calculations to be performed on incoming data. An Expression builder is available to aid in creating expressions (Figure 10-8). An expression is a data connection with the string “x=” as a token. The addition operator is able to chain the strings together.

To use a Tag within the expression, use the following syntax:
`{{Tag_name}}`

To use a local variable within the expression, use the following syntax:
`~local_variable_name~`

Types of expressions are listed in Table 10-3.

Table 10-3: Expressions

Expressions	Options
Arithmetic	addition, subtraction, multiplication, division, modules, open parenthesis, close parenthesis
Relational	less than, greater than, less than or equal, greater than or equal, equal to, not equal to
Logical	and, or, not
Bitwise	and, or, not, XOR, shift left (shl), shift right (shr)
Functional	conditional, sine, cosine, tangent, arcsine, arccosine, arctangent, square root, raised to power, logarithm, natural logarithm, exponential, absolute value, integer ceiling, integer floor, minimum, maximum

Note: A problem occurs in M-Graphics Release 3.0 and Release 4.0 when using the addition operator, in which the + adds numbers, but combines strings. Graphics created prior to M-Graphics Release 3.0 that use the addition operator may require application of a workaround. Workaround: Multiply the tag by one to force a conversion to a numeric variable type.

Like Function

The Expression Editor supports the Like function. This function performs a wildcard string compare, which searches a given string for a certain pattern/string. Table 10-4 is an overview of the Like function.

Table 10-4: Like Function

Symbol	Description	Example
Like	Wildcard string compare	Like (*string, pattern, case sensitive)

*String is the string to search, pattern is the string to search for (includes wildcards), case sensitive is nonzero for case sensitive search and zero for case sensitive search. The string syntax is \$"string"\$.

Table 10-5: Expression Examples

Expression Example	Description
<code>x= ~~A~~</code>	Local Variable Syntax
<code>x= { { <<tag>> } }</code>	Alias Syntax
<code>x= ~~A~~%~~B~~</code>	Calculates remainder after division.
<code>x= max (~~A~~,~~B~~)</code>	Returns greater of two local variables.
<code>x= ~~A~~ < ~~B~~</code>	Returns 0 if false or 1 if true.
<code>x= ~~A~~&&~~B~~</code>	Returns 1 if both are true, 0 if both are false.
<code>x= if (~~A~~>~~B~~, c, d)</code>	Returns c if true and d if false.
<code>x= pow (~~A~~, ~~B~~)</code>	Returns value A to the power of B.

Data Source Aliases

An Alias is a string (usually a short name) to represent another string (usually part or all of a Tag name). Aliases are helpful when a symbol appears in several displays with different Tags.

When including an Alias name in a data connection, the Alias name should be surrounded by the special tokens << and >>. These tokens are used to identify the beginning and end of the Alias name.

For example: <<tank>>.out. In this case, “tank”, is the Alias name.

If ‘JC.CFOPC\SupplyFan’ is used to replace the Alias, the tag becomes: JC.CFOPC\SupplyFan.out.

Object Level Aliases

Objects that utilize data connections can maintain their own Alias table. Each Dynamic object can have an Alias table. This Alias table is automatically created when you enter a data source string containing Alias names. The table is automatically populated with entries for each Alias name used in the data connections of the Dynamic object. By default, the Alias replacement string is set equal to the Alias name. If the Alias name is equal to the Alias replacement string, the Alias will not be resolved. You are only allowed to change the replacement strings in object level Alias tables. The Alias names in the table are controlled by the Alias names used in the data source strings.

To edit object level Aliases in M-Graphics, select one or more visible objects prior to executing the edit Alias table functionality. All the Alias tables for the dynamics attached to the selected objects are merged and displayed in a single Edit Aliases dialog box. If the selected objects have object names defined, those names identify which Alias names belong to which objects.

If no objects are selected when you choose to edit Aliases, all the Aliases in the display appear in the Edit Aliases dialog box. Sorting is supported in the Edit Aliases dialog box.

The Alias replacement strings are modifiable during Runtime via OLE automation.

Runtime Alias

Runtime Aliasing encompasses a variety of ways to change a display's Aliases during Runtime. The user can specify a collection of Aliases to be set as part of a given command. For example, when using a pick action to load a display, the user can specify Aliases to be set in the display that is about to be loaded. These Aliases are set before any Tags in the new display are requested. The Aliases to be set can be specified directly in M-Graphics or can come from an external tab-delimited text file. (Such files can be exported from Excel, created in Notepad, or created using M-Graphics Alias File Editor.)

Commands that Alias can be associated with include:

- Pick action-Load display
- Pick action-Drag/drop load display
- Pick action-Popup window
- Pick action-Embedded window
- Pick action-Set Aliases
- Pick action-Alias
- Tab Load Display
- Launch M-Graphics from the command line and specify an Alias file as one of the command line parameters

Procedure Overview

Table 10-6: Establishing Data Source Connections

To Do This	Follow These Steps:
Connect to a Data Source	Select an object in the display. On the Dynamics menu, select a dynamic to add. Enter the data source in the field or click on the OPC Tags button to open the OPC Universal Tag Browser. Selecting Expressions from the options of the Tags button to open the Expression Editor may also do this. Fill in the remaining dynamic fields.
Operate the OPC Universal Tag Browser	Click on My Computer in the OPC Universal Tag Browser. Navigate the tree until the point desired is in the list box. Click OK.
Add a Simulated Variable	From any Property Inspector dialog box, click the Tags Menu button. Select Simulation Variable. Select a simulated value and click OK.
Edit Expressions	In the Property Inspector, click Expression to open the Edit Expressions dialog box. Edit the fields. Click OK.
Edit Local Variables	On the Dynamic menu, select Edit Local Variables. Edit the fields. Click OK.
Create an Alias	Select an object. On the Dynamics menu, select a dynamic to apply. In the Data Source field, enter an Alias. Edit the Alias.
Edit an Alias	Select an object. On the Dynamics menu, select Edit Aliases. Fill in the fields. Click OK.
Set Runtime Aliases	Click on the Set Aliases button on the Action tab in the Property Inspector. Fill in parameters.

Detailed Procedures

Connecting to a Data Source

To connect to a data source:

1. Select an object in the display.
2. On the Dynamics menu, select a dynamic to add. The Property Inspector appears for the type of dynamic specified. Figure 10-5 is an example of a Rotation dynamic.

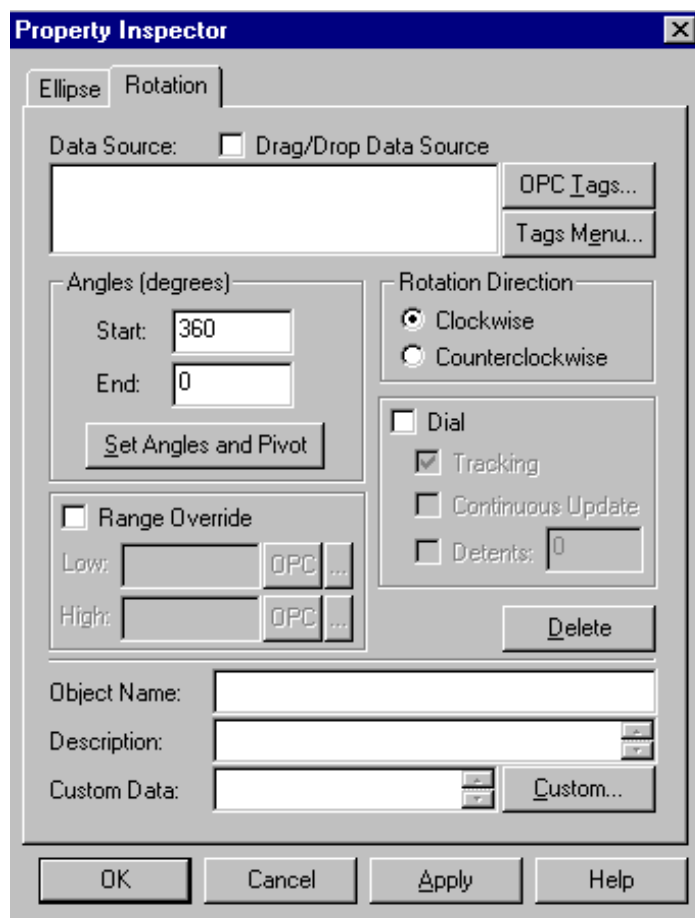


Figure 10-5: Property Inspector for a Rotation Dynamic

3. Enter the data source in the field or click on the OPC Tags button to open the OPC Universal Tag Browser. Refer to the *Operating the OPC Universal Tag Browser* section in this chapter for details.

Note: Clicking on the Tags Menu button to open the Edit Expressions or Simulated Variables is another way to perform step three. Refer to the *Editing Expressions* section in this chapter for details.

4. Fill in the remaining dynamic fields. Refer to the *Adding Dynamics* chapter of this manual.

Operating the OPC Universal Tag Browser

To operate the OPC Universal Tag Browser:

1. Click on My Computer in the OPC Universal Tag Browser and then OPC Data Access. This expands to give you a listing of OPC Data Access servers (Figure 10-6). Table 10-7 explains the OPC Universal Tag Browser fields.

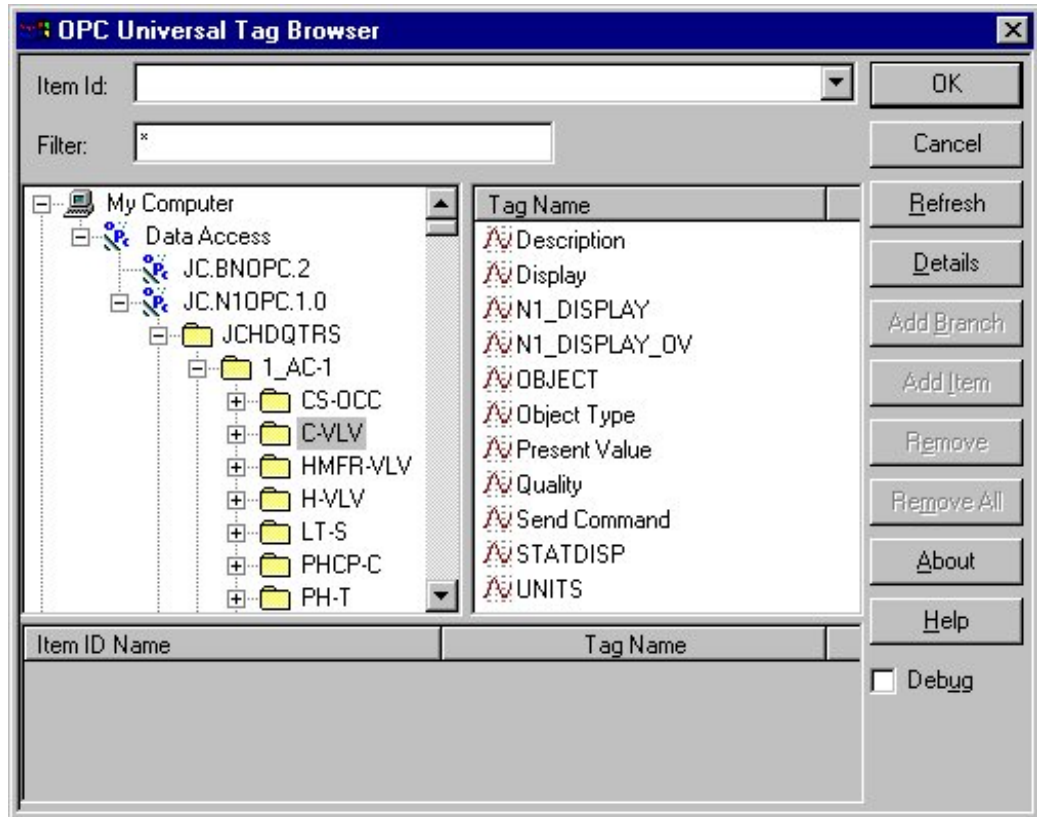


Figure 10-6: OPC Universal Tag Browser Dialog Box

Table 10-7: Areas of the OPC Universal Tag Browser

Areas of the OPC Universal Tag Browser	Description
Item ID	Shows the full Tag name. A list of previously selected points is available in the drop-down list.
Filter	Sends a filter to the OPC server.
Tag Name	Displays the full Tag name in Item ID box if you double-click or highlight Tag Names list.
Refresh	Displays updated network information.
Details	Displays information about specific item.
Add Branch	Allows an additional branch to be added to the tree diagram.
Add Item	Allows an additional item to be added.
Remove	Allows selected item to be removed.
Remove All	Allows everything in the Tag Name to be deleted.
Debug	Debugs selected Tag when checked.

2. Navigate the tree until the point desired is in the Tree list box. Double-click on the Tag Name in the Tag name list box or highlight the name in the Tag Name list.

Notes: Alternatively, select a point previously configured in the drop-down list or type in the entire name in the following format: \\node name\server name\item id.

For local server, type in: server name\item id.

3. Click OK to finish point configuration.

Enter a string in the filter field to send a filter to the OPC Data server (only available on OPC Data servers that support filtering).

Adding a Simulated Variable

To add a simulated variable:

1. From any Property Inspector dialog box, click the Tags menu button (Figure 10-1).
2. Select Simulation Variable. The Simulated Values dialog box appears (Figure 10-7).

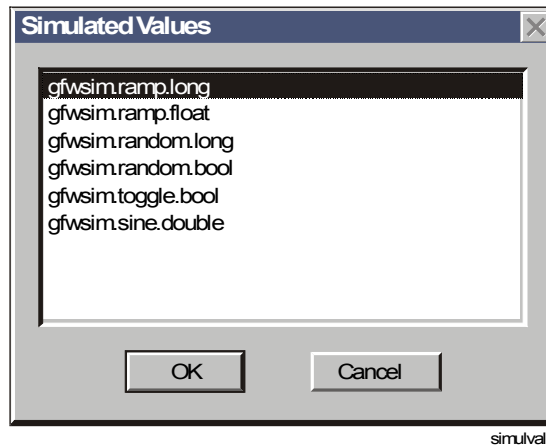


Figure 10-7: M-Graphics Simulated Values Dialog Box

3. Select a simulated value and click OK.

Editing Expressions

To edit expressions:

1. In the Property Inspector, click Tags menu to open the Edit Expression dialog box (Figure 10-8).
2. Create or edit expressions. An example is:
 $((\{\{<<\text{cctag1}>>\}) + \{\{<<\text{cctag2}>>\}) / 2.0$ is the average of two analog values.
3. Click OK.

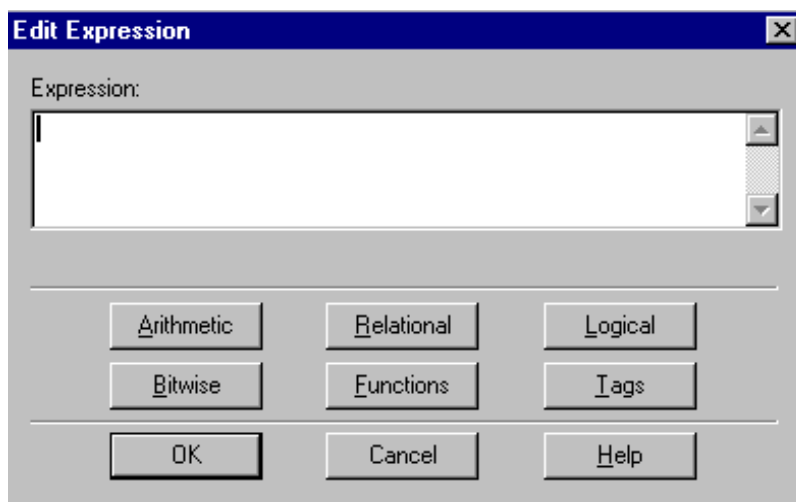


Figure 10-8: Edit Expression Dialog Box

Editing Local Variables

To edit local variables:

1. On the Dynamic menu, select Edit Local Variables. The Edit Local Variables dialog box appears (Figure 10-9).

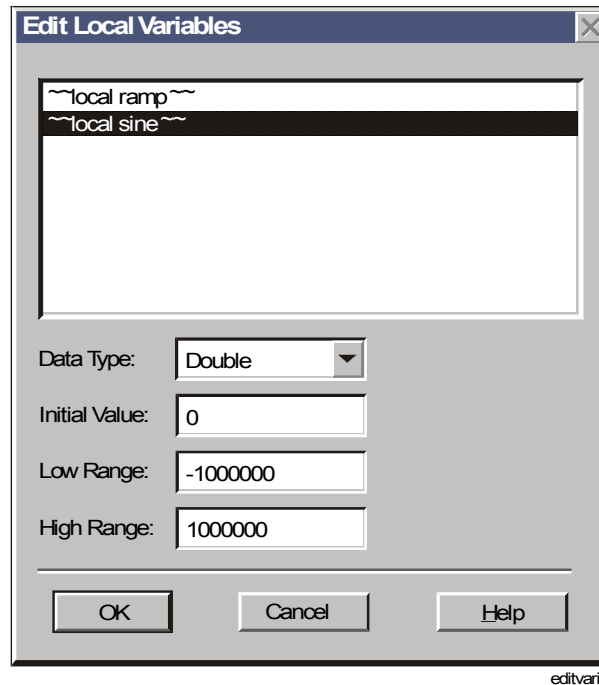


Figure 10-9: Edit Local Variables Dialog Box

2. Edit the fields using Table 10-8.
3. Click OK.

Table 10-8: Edit Local Variables Parameters

Parameter	Description
Data Type	Specifies the data type of the local variable.
Initial Value	Sets the initial value of the local variable.
High Range and Low Range	Sets a range of values for the local variable.

Creating an Alias

To create an Alias:

1. Select an object.
2. On the Dynamics menu, select a dynamic to apply.
3. In the Data Source field, enter an Alias using the following syntax: <<FANSPEED>>.
4. Edit the Alias. Refer to the *Editing an Alias* section.

Note: In M-Graphics Release 4.0, an Alias can be used to refer to another Alias.

Editing an Alias

To edit an Alias:

1. Select an object. If no objects are selected, all the Aliases for the display appear.
2. On the Dynamics menu, select Edit Aliases. The Edit Aliases dialog box appears (Figure 10-10).

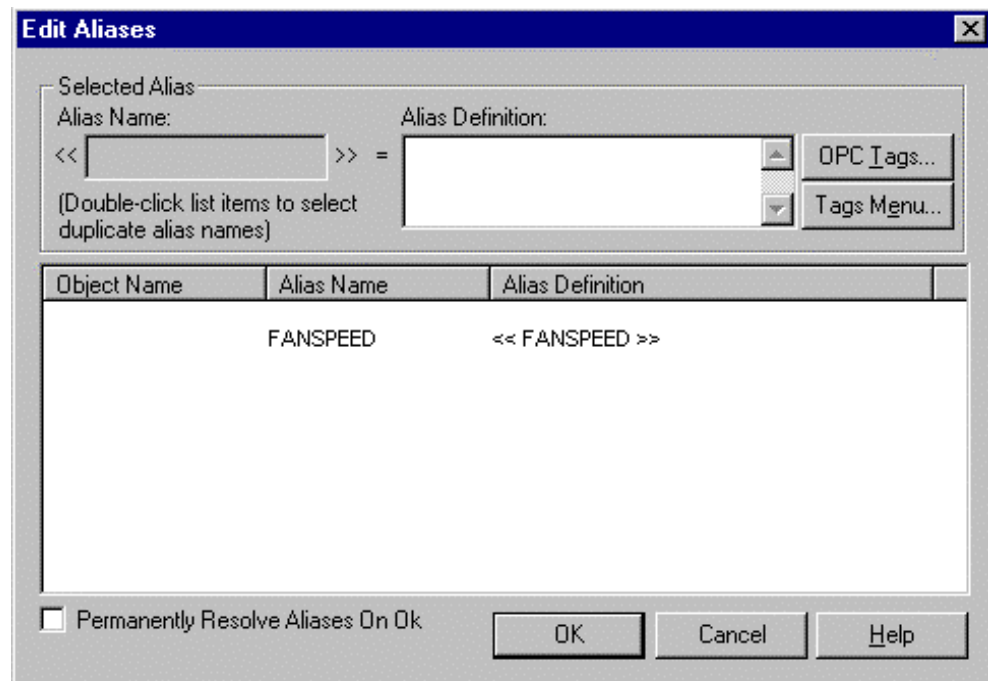


Figure 10-10: Edit Aliases Dialog Box

3. Fill in the fields following Table 10-9.

Table 10-9: Edit Aliases Preferences

Field	Description
Alias Name	Displays the name of the current object selected.
Alias Definition	Changes the definition for all selected Alias names.
OPC Tags	Displays the OPC Universal Tag Browser dialog box.
Tags Menu	Opens the Expression Editor, Alias, Local Variables or Simulation Variables dialog box.
Permanently Resolved Aliases On Ok	Resolves Aliases when the OK button is clicked. Aliases are normally resolved when entering Runtime mode. Resolves the Aliases in Configuration mode.

4. Repeat Step 3 for all Aliases.
5. Click OK.

Setting Runtime Alias

To set Runtime Alias:

1. Click the Set Aliases button on the Pick tab of the Property Inspector. The Set Aliases Configuration dialog box appears (Figure 10-11).

Note: This interface may vary slightly depending upon the action being configured.

2. Fill in parameters using Table 10-10.

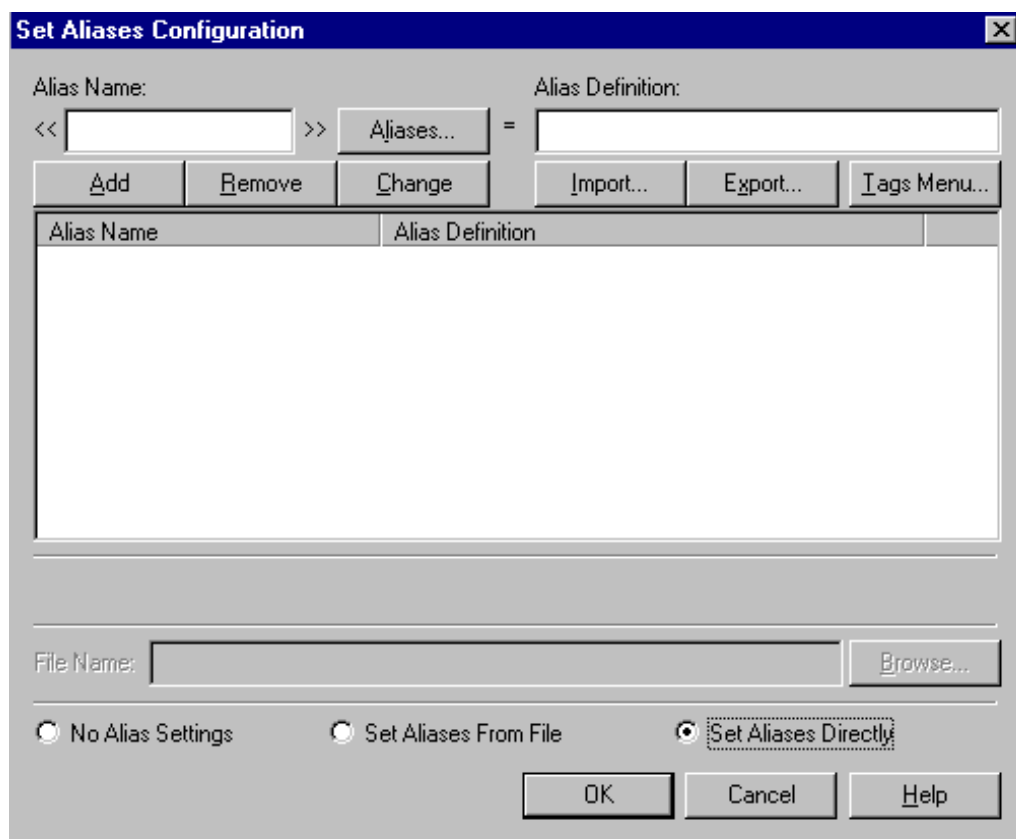


Figure 10-11: Set Aliases Configuration Dialog Box

Table 10-10: Set Aliases Configuration Dialog Box Parameters

Parameter	Description
Alias Name	Shows Name assigned to Alias.
Aliases	Displays definitions for all Aliases.
Alias Definition	Changes the definitions for all selected Alias names.
Add	Adds Alias to list.
Remove	Removes Alias from list.
Change	Changes an Alias that is currently assigned.
Import	Imports an Alias to the list.
Export	Exports an Alias from the list.
Tags Menu	Displays the Edit Expressions, Alias, Local Variables or Simulated Variables dialog box.
File Name	Contains the name of the text file. Use the Browse button to locate a file.
No Alias Settings	If this option is selected, no Aliases can be specified.
Set Aliases From File	The Aliases can come from an external tab-delimited text file. These files can be exported from Excel, created in Notepad, or created using M-Graphics Alias File Editor.
Set Aliases Directly	The Aliases can be specified directly in M-Graphics and saved.

The Alias replacement strings are modifiable during Runtime via OLE automation.

Adding Dynamics

Introduction

This chapter explains how to make single or multiple dynamic connections from display objects to points from OPC data servers. This chapter describes how to:

- add a size dynamic
- add a location dynamic
- add a rotation dynamic
- add a hide dynamic
- add a color dynamic
- add an analog color dynamic
- add a flash dynamic
- add a pick dynamic
- add a digital selector dynamic
- add an analog selector dynamic
- add an animator dynamic
- add time/date
- create a Process Point/Data Entry (PPT/DE)
- create a pushbutton
- create a checkbox
- create a radio button
- create a display button

Key Concepts

Gradient Fill Feature

The Gradient Fill feature supports the display background, rectangles and rounded rectangles, ellipses, pies and chords, lines, and text-back fill (including Process Point [PPT]), buttons, and timedates). To access this feature, insert one of the items mentioned above and double-click to open the Property Inspector.

Gradient Configuration Features

Object colors may be changed from the Configure Gradient dialog box (Figure 11-1). Clicking on the desired color opens the Color Palette. To reverse the color, when using both single and two color gradients, check the Reverse Colors checkbox on the Configure Gradient dialog box.

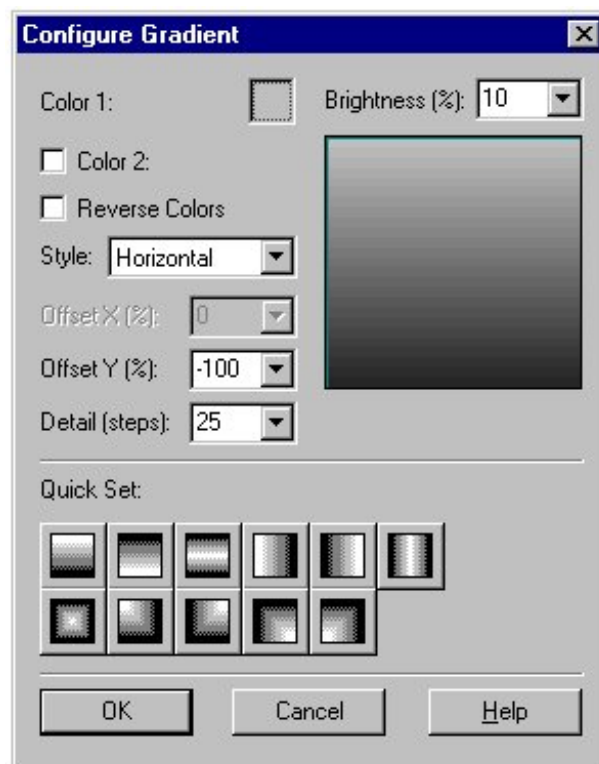


Figure 11-1: Configure Gradient Dialog Box

Gradient Styles

There are three different gradient styles: Horizontal, Square, and Vertical. Table 11-1 describes the different styles. All three styles are supported on Microsoft® Windows® 2000 Professional and Windows XP Professional operating systems.

Table 11-1: Gradient Styles

Style	Description
Horizontal	The Horizontal style has the light focus coming from the top or the bottom of the object.
Square	The Square style has the light focus coming from a corner.
Vertical	The Vertical style has the light focus coming from either the right or the left side of the object.

Gradient Offsets

This feature allows the variation of colors to be selected. There are two options available in the Configure Gradient dialog box: Offset X and Offset Y. Figure 11-2 gives an example of these options.

Note: Any number between 100 and -100 may be entered for both the Offsets.

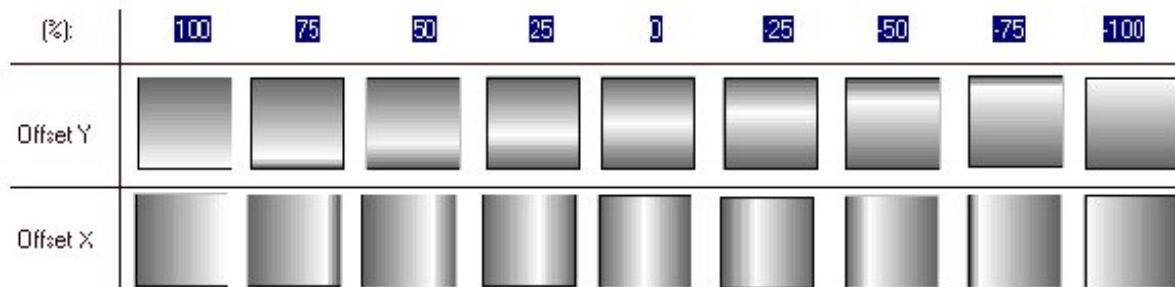


Figure 11-2: Offset X and Offset Y by Percentage

Brightness

When only using one color, the option of selecting a percentage of brightness is available. Any number between 0 and 100 may be entered for the brightness level.

When using two colors, the brightness adjustment option disappears and a second color may be selected.

Dynamics

Dynamics animate and control display objects using values established in the data source connection. Refer to the *Establishing Data Source Connections* chapter for details.

Add multiple dynamic connections to a single object. For example, you can create an object that changes color and size based on values from point connections.

Types of dynamic connections include action, selector, and intrinsic.

Action Dynamics

When an Action Dynamic is added to an object that has a gradient, the object acts as it would normally. The coloration simply contains the gradient fill. If the object is given a rotation dynamic, the shading of the gradient changes in areas. For example, as the object rotates past the X-axis, the shading flips sides of the object.

Several action dynamics are discussed in the following sections.

Size

Use the size dynamic to define an object that changes size based on the realtime value of its connected data point. The object automatically sizes to scale for incoming data points, such that a 100% scale equals the full size of the drawn object. Override the size by defining range override values.

During Runtime, the size dynamic resizes the object in proportion to the percentage of the high/low range of the connected data point.

Use the size dynamic for bar graphs, liquid levels, and thermometer mercury.

Location/Slider

The location dynamic changes the location of an object based on the value of a data point. During Runtime mode, the object moves within a specified distance—horizontally, vertically, diagonally, or along a specified path—based on the value of the connected data point.

The slider dynamic emulates a mechanical slider control. During Runtime mode, move the slider knob to change the value of the connected data point.

Use the location/slider dynamic for arrows, slider bars, and motion through pipes.

Rotation/Dial

The rotation/dial dynamic rotates an object based on the connected data point value. You define the rotation pivot coordinates. This function is useful when graphically displaying information on dials or meters in the display.

Use the rotation/dial dynamic for dampers, meters, gauges, and levers.

Hide/Disable

The hide/disable dynamic causes objects to hide when the value of the digital data point is connected to goes either true (logical 1) or false (logical 0).

Use the hide/disable dynamic for displaying information when a Boolean tag becomes true or false.

Color

The color dynamic changes the color of the object based on an event. The color connections are prioritized in the order in which they are created. Therefore, if two data points are true at the same time, the higher priority color connection takes precedence. If none of the connected data points is true, the object displays in its original color.

The color dynamic is useful for indicating such states as alarm conditions and temperature changes or for notifying operators of the flow in a pipe.

Analog Color

The analog color dynamic changes the start color and end color based on an analog signal.

Attach multiple analog color dynamics to shift between more than two colors.

Use analog color for showing approximate measurement such as temperature.

Flash

The flash dynamic causes an object with a digital connection to blink on and off or to flash between two colors during Runtime mode when the connected digital value goes true (logical 1) or false (logical 0).

The default operation is for the object to flash between configured colors. The Property Inspector shows the default as on/off. When the data point value is false (logical 0), the object displays in the normally on/normally off setting. When the value is true (logical 1), the object blinks between the colors or between visible and hidden. The rate at which the object blinks is determined by the Flash Rate value.

Pick

The pick action dynamic makes an object perform an action when clicked on during Runtime mode. The pick action options are listed in Table 11-2. To update the present value of a command for an object, enter the desired numeric value, such as 70.2, in the Value field of the Property Inspector. To enter a string command, use the following syntax:

\$“string”\$

Examples of sending string commands for a download value pick action:

To release an override command for N1 objects using the n1_display_ov attribute, enter \$“Auto”\$ in the Value field of the Property Inspector.

To command a Binary Data (BD) object at Priority 2 using a JC-BASIC TELL statement, enter \$“SET_BD 1,2”\$ in the Value field of the Property Inspector.

To display system names using the alias feature, enter \$“<<system>>”\$ in the Value field of the Property Inspector, where system is an alias name.

Table 11-2: Pick Action Options

Pick Action	Description
Load Display	Loads a display.
Drag/Drop Load	Allows you to drag/drop a display to another M-Graphics window.
Display Back	Navigates backward through the display history. M-Graphics maintains a history of the last 50 displays.
Display Forward	Navigates forward through the display history. M-Graphics maintains a history of the last 50 displays.
Popup Window	Opens a popup window.
Embedded Window	Opens an embedded window, which is a child window of the display that launched the embedded window. The embedded window can move, scroll, and scale with the parent display.
Close Window	Closes the window.
Launch Application	Launches the application specified in the Filename field.
Download Value	Downloads values when you click on the pick object.
Toggle Value	Toggles between two specified values.
Set Aliases	Sets aliases for data connections in Runtime.
Alias Dialog	Opens the Set Aliases Configuration interface to configure aliases.
Run VBA Script	Runs the specified VBA Script.
Custom Command	Runs a custom function. This function is an executable file (.exe) or a .dll.

Selector Dynamics

Selectors display specific objects based on the value of a data connection. There are three types of selectors:

- digital selector
- analog selector
- animator

Digital Selector

The digital selector connects individual objects to corresponding digital data points. During Runtime mode, when the connected data point goes true (logical 1) or optionally false, the connected object appears on the screen.

The digital selector displays one of a number of objects based on the state of digital signals. One signal is connected to each object. The highest priority signal (determined by position in the list) displays the corresponding object when true (or optionally false).

Generally, no objects appear when all signals are false (logical 0). The digital selector allows a point value to be true or false to determine which selected group of objects to view.

Analog Selector

The analog selector defines a group of objects attached to an analog data point. During Runtime mode, the value of the analog data point determines which assigned object is shown.

The analog selector displays one of a number of objects based on the value of the analog signal. This object displays when the signal falls within the range of values specified for the object.

The analog selector allows a percentage value from an analog point value to determine which selected group of symbols to view.

Animator

The animator allows selected objects to be displayed, sequentially, based on a value's being true or false.

The animator allows a group of objects to be displayed, sequentially based on the state of a digital variable to which they are connected. When that digital variable goes to its true state (logical 1) or optionally false during Runtime mode, the assigned objects appear on screen in the sequence in which they are ordered. The sequence repeats until the connected process variable returns to a false state (logical 0) or optionally true. The rate of animation is selectable.

Intrinsic Dynamics

Intrinsic dynamics creates operator controls in display files. Intrinsic dynamics make dynamic connections to control objects such as data entry objects and sliders and pushbuttons that display realtime data during Runtime mode.

Process Point/Data Entry

The Process Point/Data Entry (PPT/DE) dynamic creates an object used to display and enter data in an alphanumeric format.

A PPT is an object that displays the realtime value of a connected data point during Runtime mode. This value is updated when the server informs M-Graphics of a change in the data point value.

A DE behaves the same way as a process point and also allows you to manually enter new values to be downloaded to the system.

Time/Date

The time and date dynamic adds the current time and date to the display. This functionality supports dates beyond the year 2000. The time and date fields have Free-Format capabilities.

Pushbutton

The pushbutton dynamic creates a pushbutton in a display. The pushbutton object behaves similarly to the pick dynamic. However, the pushbuttons also animate the up/down state when selected in Runtime mode.

Checkbox

The checkbox dynamic creates a checkbox in the display. Checkboxes look and act like standard Windows checkbox controls. When selected in Runtime mode, the checkbox remains selected until it is selected again.

Radio Button

The radio button dynamic creates a radio button like a pushbutton. Radio buttons look and act like standard Windows operating system radio button controls. When selected in Runtime mode, the radio button automatically deselects all other radio buttons that belong to the same group (i.e., the radio buttons in a given group are mutually exclusive).

Display Button

A display button, once configured through a wizard, can do one of four things: load a new display, open a display in an embedded M-Graphics window, display a pop-up window, or serve as a drag/drop object that can be placed in an M-Graphics ActiveX® container to load a display. This feature reduces clutter in a display, as embedded and pop-up windows are shown only when needed by the push of a button.

Procedure Overview

Table 11-3: Adding Dynamic Connections

To Do This	Follow These Steps:
Add a Size Dynamic	Select the object. On the Dynamics menu, select Actions > Size. Establish a data connection. Fill in the size dynamic parameters. Click OK.
Add a Location Dynamic	Select the object. On the Dynamics menu, select Actions > Location. Establish a data connection. Fill in the location dynamic parameters. Click OK.
Add a Rotation Dynamic	Select the object. On the Dynamics menu, select Actions > Rotation. Establish a data connection. Fill in the rotation dynamic parameters. Click OK.
Add a Hide Dynamic	Select the object. On the Dynamics menu, select Actions > Hide. Establish a data connection. Fill in the hide dynamic parameters. Click OK.
Add a Color Dynamic	Select the object. On the Dynamics menu, select Actions > Color. Establish a data connection. Fill in the color dynamic parameters. Click OK.
Add an Analog Color Dynamic	Select the object. On the Dynamics menu, select Actions > Analog Color. Establish a data connection. Fill in the analog color dynamic parameters. Click OK.
Add a Flash Dynamic	Select the object. On the Dynamics menu, select Actions > Flash. Establish a data connection. Fill in the flash dynamic parameters. Click OK.
Add a Pick Dynamic	Select the object. On the Dynamics menu, select Actions > Pick. Establish a data connection. Fill in the pick dynamic parameters. Click OK.
Add a Digital Selector Dynamic	Select the object. On the Dynamics menu, select Selectors > Digital Selector. Establish a data connection. Fill in the digital selector dynamic parameters. Click OK.
Add an Analog Selector Dynamic	Select the object. On the Dynamics menu, select Selectors > Analog Selector. Establish a data connection. Fill in the analog selector dynamic parameters. Click OK.
Add an Animator Dynamic	Select the object. On the Dynamics menu, select Selectors > Animator. Establish a data connection. Fill in the animator dynamic parameters. Click OK.
Add Time/Date	Select the object. On the Dynamics menu, select Intrinsic > Time/Date. Click in the work area. Select a time and date format. Click OK.
Create a Process Point/Data Entry (PPT/DE)	On the Dynamics menu, select Intrinsic > Process Point. Click in the work area. Establish a data point connection. Fill in the PPT/DE dynamic parameters. Click OK.
Create a Pushbutton	On the Dynamics menu, select Intrinsic > Pushbutton. Click in the work area. The Property Inspector appears with tabs for Pick and Button. Fill in the parameters. Click OK.
Create a Checkbox	On the Dynamics menu, select Intrinsic > Checkbox. Click in the work area. The Property Inspector appears with tabs for Pick and Checkbox. Fill in the parameters. Click OK.
Create a Radio Button	On the Dynamics menu, select Intrinsic > Radio Button. Click in the work area. The Property Inspector appears with tabs for Pick and Radio Button. Fill in the parameters. Click OK.
Create a Display Button	On the Dynamics menu, select Intrinsic > Display Button Wizard. Click in the work area. The Display Button Wizard appears. Fill in parameters. Click OK.

Detailed Procedures

Adding a Size Dynamic

To add a size dynamic:

1. Select the object.
2. On the Dynamics menu, select Actions > Size. The Property Inspector appears with the Size tab (Figure 11-3).

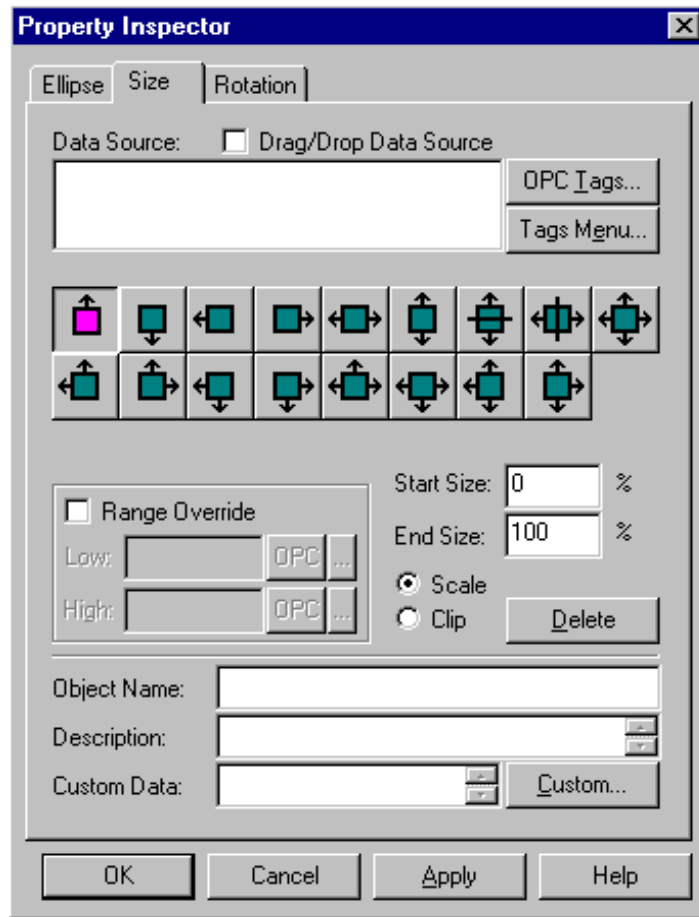


Figure 11-3: Property Inspector Dialog Box - Size Tab

3. Establish a data connection.
4. Fill in the size dynamic parameters using Table 11-4.
5. Click OK.

Table 11-4: Size Action Parameters

Parameter	Description
Data Source	Lists the tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a Tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Size Toolbar	Specifies the direction to size the object.
Range Override	Activates an operating range other than the default range for the data point.
Start Size/End Size	Specifies the starting and ending size of the object that has changing dimensions.
Scale	Scales the object in proportion to the incoming signal.
Clip	Reveals the object.
Delete	Deletes this dynamic.
Object Name	Identifies the object for OLE automation.
Description	Describes the object and determines the ToolTips displayed.
Custom Data	Allows user to enter data.

Adding a Location Dynamic

To add a location dynamic:

1. Select the object.
2. On the Dynamics menu, select Actions > Location. The Property Inspector appears with the Location tab (Figure 11-4).

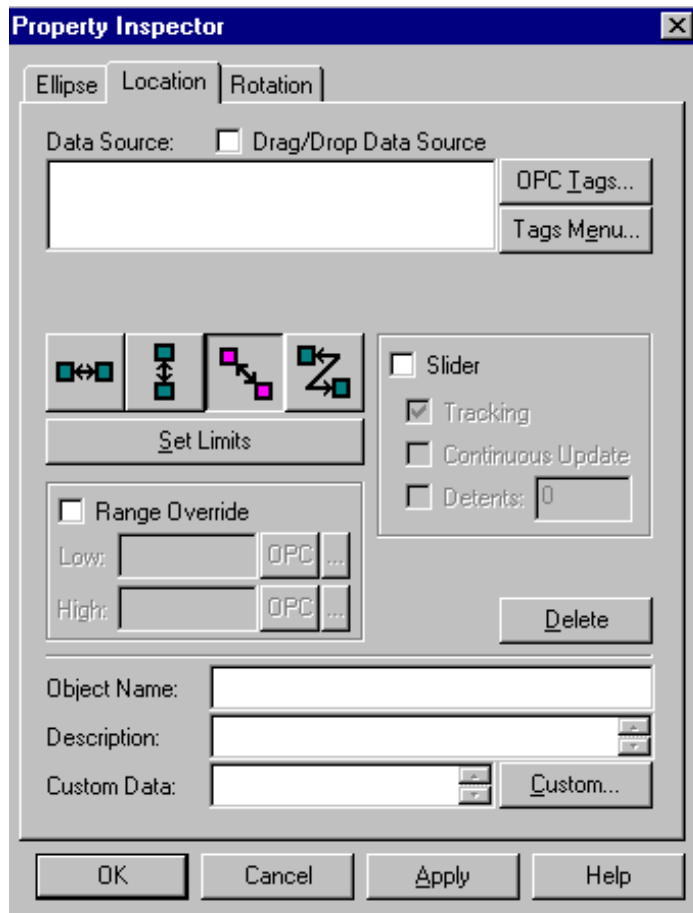


Figure 11-4: Property Inspector Dialog Box – Location Tab

3. Establish a data connection.
4. Fill in the location dynamic parameters using Table 11-5.
5. Click OK.

Table 11-5: Location Parameters

Parameter	Description
Data Source	Lists the tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Direction Buttons	Determines object direction: side to side, up and down, back and forth diagonally, or free-form.
Range Override	Activates an operating range other than the default for the data point.
Set Limits	Allows range of motion limits to be defined graphically.
Slider	Allows you to hold an object with the mouse and move it.
Tracking	Toggles if the object moves based on the connected data value or remains stationary.
Continuous Update	Toggles if the dynamic is continuously updated.
Detents	Moves the object at specific intervals.
Object Name	Identifies the object for OLE automation.
Description	Describes the object and determines the ToolTips displayed.
Custom Data	Allows user to enter data.
Delete	Deletes this dynamic.

Adding a Rotation Dynamic

To add a rotation dynamic:

1. Select the object.
2. On the Dynamics menu, select Actions > Rotation. The Property Inspector dialog box appears with the Rotation tab Figure 11-5.

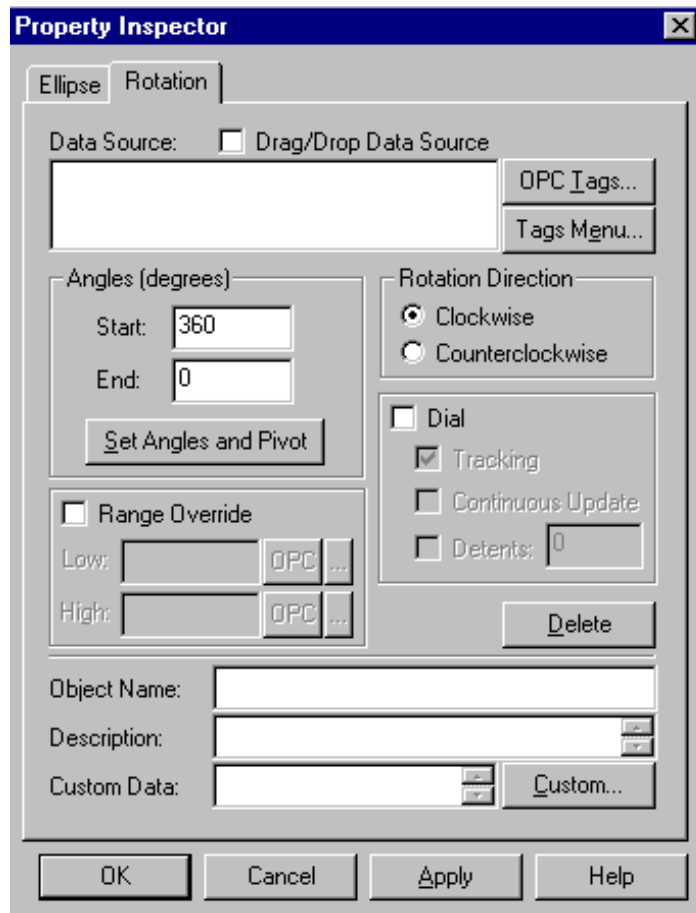


Figure 11-5: Property Inspector Dialog Box – Rotation Tab

3. Establish a data connection.
4. Fill in the rotation dynamic parameters using Table 11-6.
5. Click OK.

Table 11-6: Rotation Parameters

Parameter	Description
Data Source	Lists the tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Start	Defines the angle to start the rotation.
End	Defines the angle to end the rotation.
Set Angles and Pivot	Allows graphically setting the angle of rotation and defines the coordinates of the pivot point of the object.
Rotation Direction	Defines a rotation direction, clockwise or counterclockwise.
Dial	Allows you to manually rotate the object.
Tracking	If checked, moves the object based on the connected data value. If unchecked, the object remains stationary unless manually moved by you. This is part of the dial action.
Continuous Update	Toggles if the dynamic is continuously updated.
Detents	Moves the object in steps at specific intervals. This is part of the dial action.
Range Override	Activates an operating range other than the default range for the data point.
Delete	Deletes this dynamic.
Object Name	Identifies the object for OLE automation.
Description	Describes the object and determines the ToolTips displayed.
Custom Data	Allows user to enter data.

Adding a Hide Dynamic

To add a hide dynamic:

1. Select the object.
2. On the Dynamics menu, select Actions > Hide. The Property Inspector appears with the Hide tab (Figure 11-6).

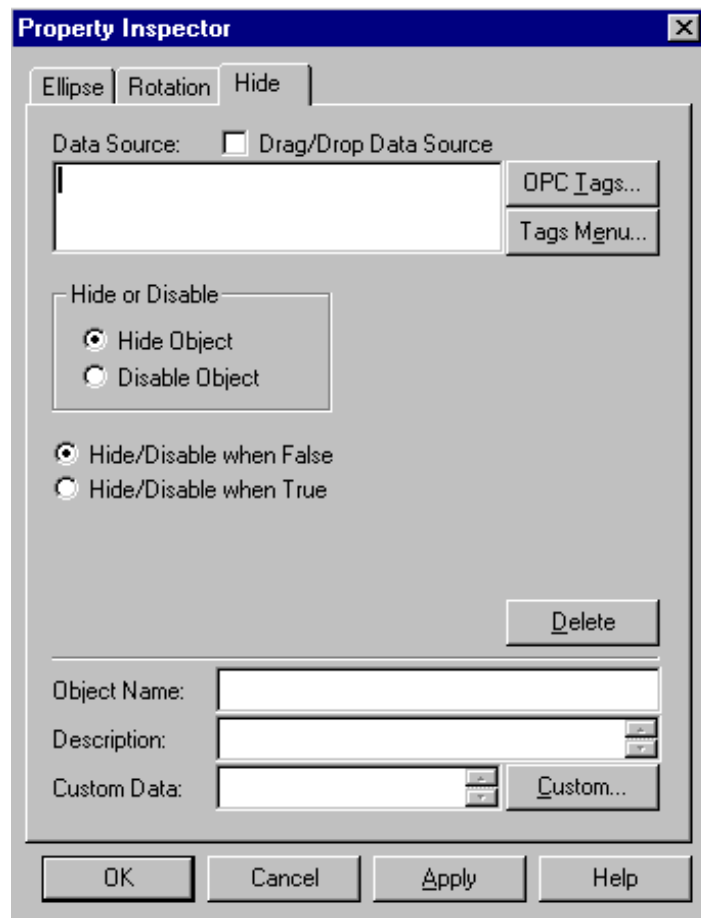


Figure 11-6: Property Inspector Dialog Box - Hide Tab

3. Establish a data connection.
4. Select Hide Object or Disable Object.
5. Select the Hide/Disable when False or Hide/Disable when True.
6. Click OK.

Table 11-7: Hide Parameters

Parameter	Description
Data Source	Lists the tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Hide or Disable	Specifies whether to hide or disable object.
Hide/Disable when False	Status of Data Source is False when object is hidden or disabled.
Hide/Disable when True	Status of Data Source is True when object is hidden or disabled.
Delete	Deletes this dynamic.
Object Name	Identifies the object for OLE automation.
Description	Describes the object and determines the ToolTips displayed.
Custom Data	Allows user to enter data.

Adding a Color Dynamic

To add a color dynamic:

1. Select the object.
2. On the Dynamics menu, select Actions > Color. The Property Inspector dialog box appears with the Color tab (Figure 11-7).

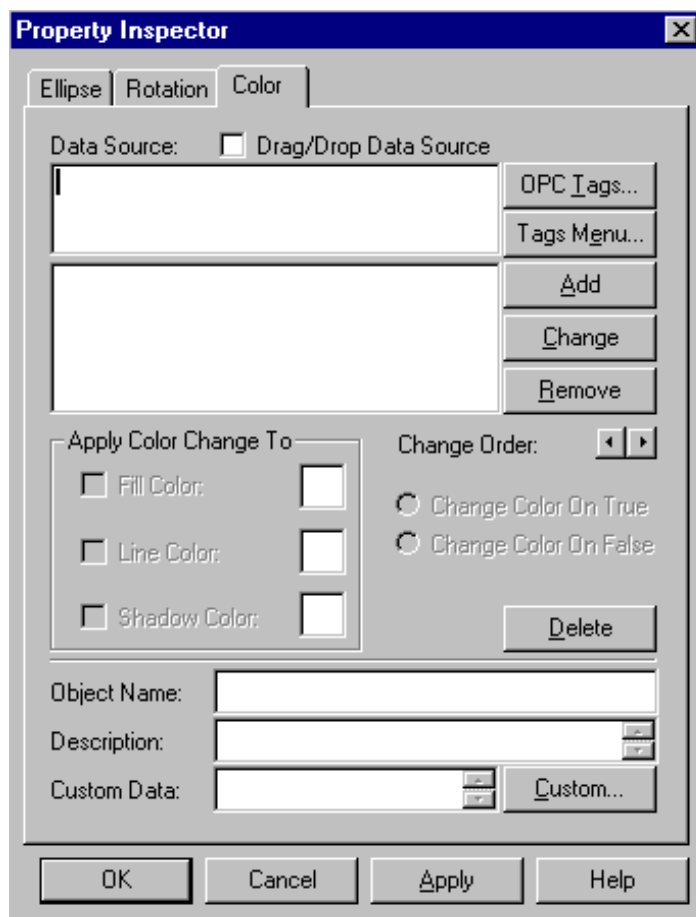


Figure 11-7: Property Inspector Dialog Box - Color Tab

3. Establish a data connection.
4. Fill in the color dynamic parameters using Table 11-8.
5. Click OK.

Table 11-8: Color Connection Parameters

Parameter	Description
Data Source	Lists the tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a Tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Add	Adds the string currently in the data source edit control to the list of data sources.
Change	Changes existing data connections. The currently selected item in the data source list box is changed to the string currently in the data source edit control.
Remove	Removes existing data connections from the data source list box.
Apply Color Change To	Varies depending on which source string is selected.
Fill Color	Changes the object's fill color.
Line Color	Changes the object's border/line color.
Shadow Color	Changes the object's shadow color.
Change Color On True/False	Changes the object's color when the connected data point is true (logical 1) or false (logical 0).
Delete	Deletes this dynamic.
Object Name	Identifies the object for OLE automation.
Description	Describes the object and determines the ToolTips displayed.
Custom Data	Allows user to enter data.

Adding an Analog Color Dynamic

To add an analog color dynamic:

1. Select the object.
2. On the Dynamics menu, select Actions > Analog Color. The Property Inspector dialog box appears with the Analog Color tab (Figure 11-8).

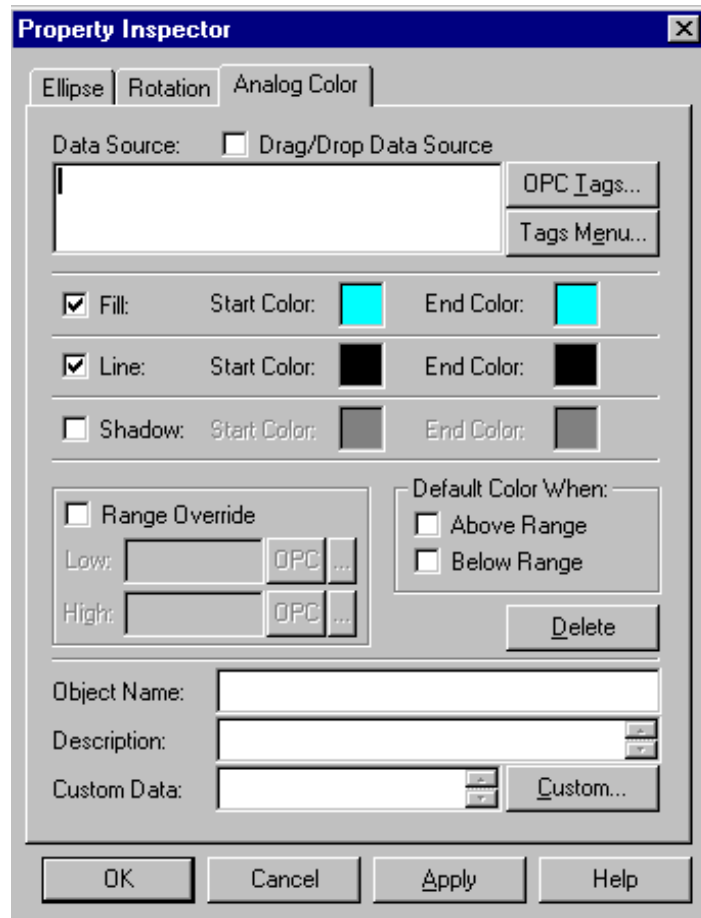


Figure 11-8: Property Inspector Dialog Box - Analog Color Tab

3. Establish a data connection.
4. Fill in the analog color parameters using Table 11-9.
5. Click OK.

Table 11-9: Analog Color Parameters

Parameters	Description
Data Source	Lists the Tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a Tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Fill Color - Start/End	Changes starting and ending fill color of the object.
Start Color	Changes starting fill color of the object when the analog signal is at its lowest value.
End Color	Changes ending fill color of the object. This is the color that appears when the analog signal is at its highest value. When the analog signal is between its minimum and maximum values, the color is a mix of the start and end colors.
Line Color - Start/End	Changes starting and ending line color of the object.
Shadow Color - Start/End	Changes starting and ending shadow color of the object.
Range Override - Low/High	Activates an operating range other than the default with specific options of choosing low and/or high range.
Default Color When Above/Below Range	Sets default color when above or below the specified range.
Delete	Deletes this dynamic.
Object Name	Identifies the object for OLE automation.
Description	Describes the object and determines the ToolTips displayed.
Custom Data	Allows user to enter data.

Adding a Flash Dynamic

To add a flash dynamic:

1. Select the object.
2. On the Dynamics menu, select Actions > Flash. The Property Inspector dialog box appears with the Flash tab (Figure 11-9).

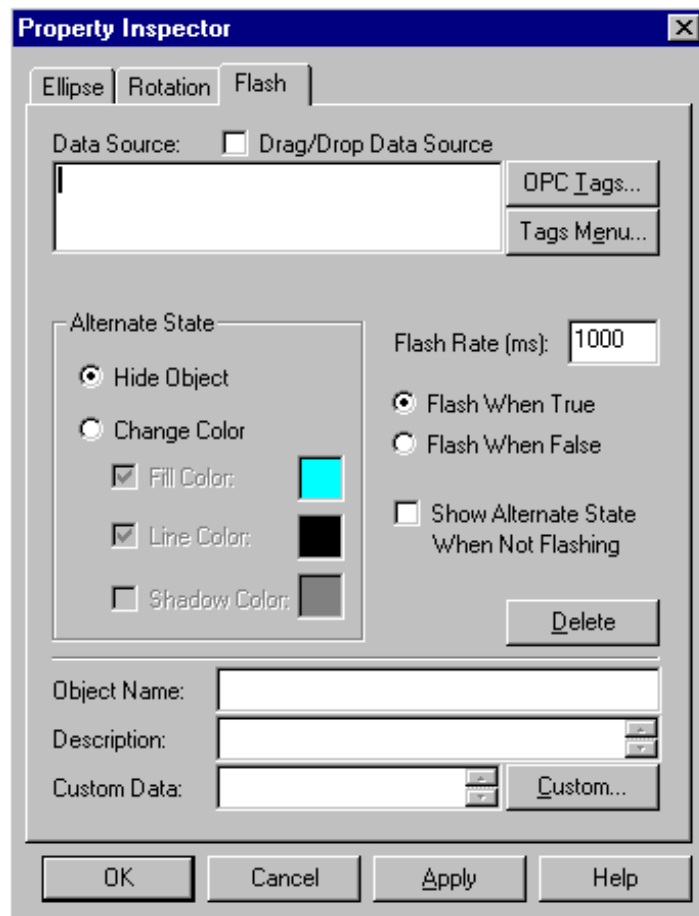


Figure 11-9: Property Inspector Dialog Box - Flash Tab

3. Establish a data connection.
4. Fill in the flash parameters using Table 11-10.
5. Click OK.

Table 11-10: Flashing Action Parameters

Parameter	Description
Data Source	Lists the Tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a Tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Flash Rate	Lists the frequency of the flash in milliseconds. The minimum is 50 ms.
Flash When True	Flashes when the value is true (logical 1).
Flash When False	Flashes when the data point value is false (logical 0).
Hide Object	Determines the alternate state is hidden object.
Change Color	Determines the alternate state is a color change.
Fill Color	Selects the object's fill color.
Line Color	Selects the object's line color.
Shadow Color	Selects the object's shadow color.
Show Alternate State When Not Flashing	Shows the object in the specified alternate color, or the object is hidden.
Delete	Deletes this dynamic.
Object Name	Identifies the object for OLE automation.
Description	Describes the object and determines the ToolTips displayed.
Custom Data	Allows user to enter data.

Adding a Pick Dynamic

To add a pick dynamic:

1. Select the object.
2. On the Dynamics menu, select Actions > Pick. The Property Inspector dialog box appears with the Pick tab (Figure 11-10).

Note: The appearance of this dialog box varies depending upon which action is selected.

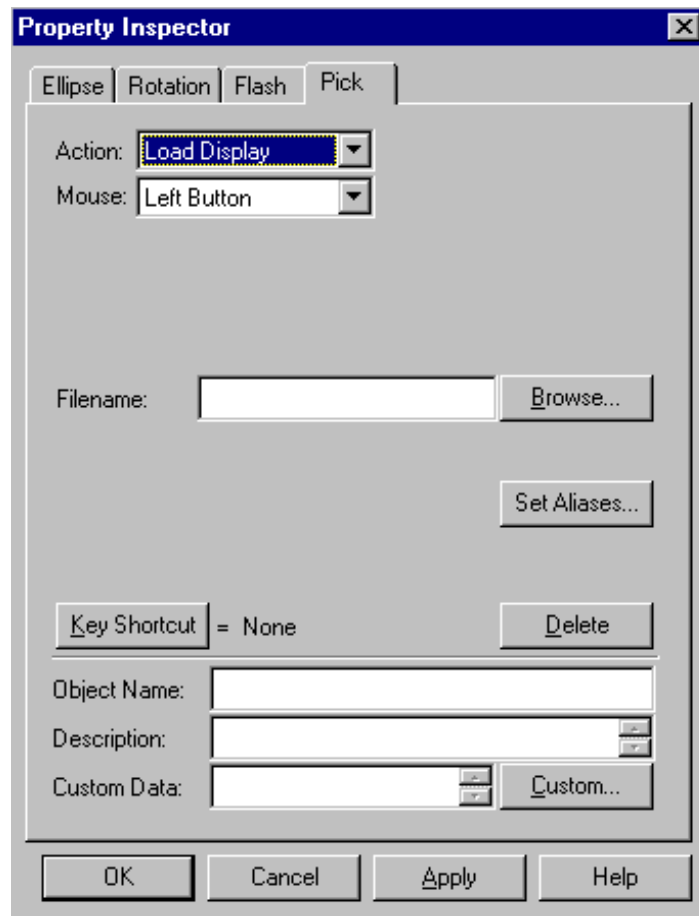


Figure 11-10: Property Inspector Dialog Box- Pick Tab

3. Establish a data connection.
4. Fill in the pick dynamic parameters using Table 11-11.

Note: Additional parameters appear depending on the type of pick action.

Table 11-11: Pick Parameters

Parameter	Description
Action	Lists all the Pick action options: load display, drag/drop load, display back, display forward, popup window, embedded window, close window, launch application, download value, toggle value, set aliases, alias dialog, run VBA Script, and custom command.
Mouse	Lists the mouse button that activates the Pick action (left, middle, or right).
Filename	Displays the name of the current file.
Set Aliases	Opens the Alias Configuration dialog box.
Key Shortcut	Defines a shortcut for this Pick action.
Delete	Deletes this dynamic.
Object Name	Identifies the object for OLE automation.
Description	Describes the object and determines the ToolTips displayed.
Custom Data	Allows user to enter data.

Adding a Digital Selector Dynamic

To add a digital selector dynamic:

1. Select two or more objects.
2. On the Dynamics menu, select Selectors > Digital Selector. The Property Inspector dialog box appears with the Digital Selector tab (Figure 11-11).

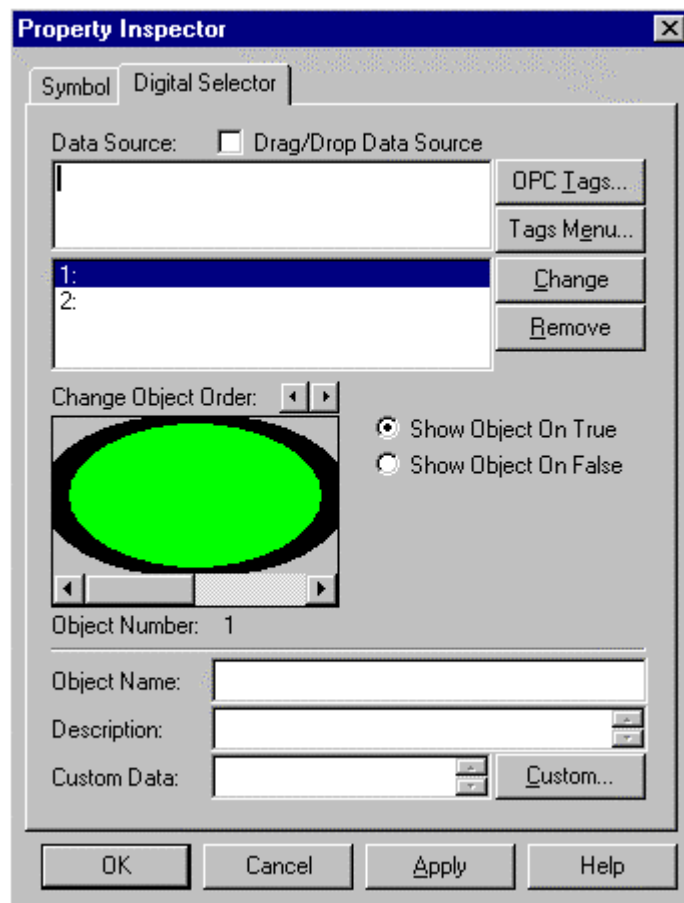


Figure 11-11: Property Inspector Dialog Box – Digital Selector Tab

3. Establish a data connection.
4. Fill in the digital selector dynamic parameters using Table 11-12.

Table 11-12: Digital Selector Parameters

Parameter	Description
Data Source	Lists the Tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a Tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens the Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Change Button	Changes existing data connections. The selected item in the data source list box is changed to the string in the data source edit control.
Remove Button	Removes currently selected data connections from the data source list box.
Object Number	Shows the objects in the order created and their data point connections assigning priorities to the objects.
Change Object Order	Changes the object order allowing you to assign new priorities for the data connections.
Show Object On True	Shows the object when the connected data point goes true (logical 1).
Show Object On False	Shows the object when the connected data point goes false (logical 0).

Adding an Analog Selector Dynamic

To add an analog selector dynamic:

1. Select two or more objects.
2. On the Dynamics menu, select Selectors > Analog Selector. The Property Inspector dialog box appears with the Analog Selector tab (Figure 11-12).

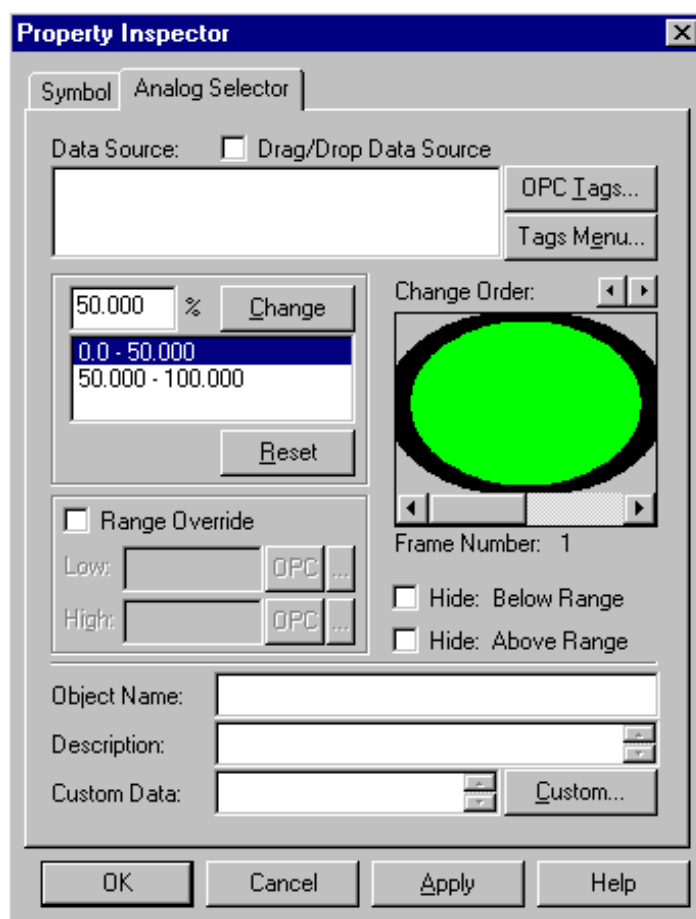


Figure 11-12: Property Inspector Dialog Box – Analog Selector Tab

3. Establish a data connection.
4. Fill in the analog selector dynamic parameters using Table 11-13.

Table 11-13: Analog Selector Parameters

Parameter	Description
Data Source	Lists the Tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a Tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens the Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Percentage	Allows objects to display when the data point value is at a certain percentage of the full range.
Percentage List	Displays the percentage for each object.
Change	Changes the percentage chosen if you type a new value in the percentage field.
Reset	Resets the percentages chosen for the items to the default values (the default values are an even distribution within the available range based on the number of objects selected).
Change Order	Change the order of the objects belonging to this analog selector.
Range Override	Activates an operating range other than the default range for the data point when this box is checked. Allows you to specify a High Range and a Low Range. Provides the Tags and Expression button to help you do so.
Frame Number	Shows the objects in the order they were created.
Hide Below Range	Hides all the associated objects when the value of the connected data point is below the specified range.
Hide Above Range	Hides all the associated objects when the value of the connected data point is above the specified range.

Adding an Animator Dynamic

To add an animator dynamic:

1. Select two or more objects.
2. On the Dynamics menu, select Actions > Animator. The Property Inspector dialog box appears with the Animator tab (Figure 11-13).

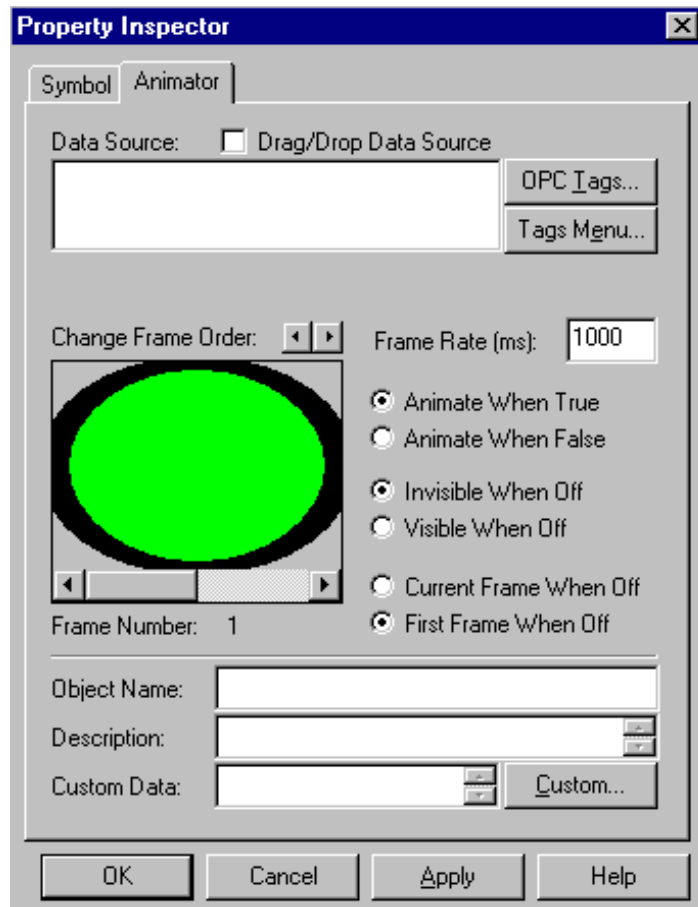


Figure 11-13: Property Inspector Dialog Box - Animator Tab

3. Establish a data connection.
4. Fill in the animator dynamic parameters using Table 11-14.
5. Click OK.

Table 11-14: Animator Dynamic

Parameters	Description
Data Source	Lists the Tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a Tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens the Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Change Frame Order	Changes the sequence number of the currently displayed object. Allows you to reorder the object.
Frame Number	Displays sequence number of currently displayed object.
Frame Rate (ms)	Lists the rate (in milliseconds) the frames change while animating.
Animate When True	Animates the object when the digital variable goes to its true state (logical 1).
Animate When False	Animates the object when the digital variable goes to its false state (logical 0).
Invisible When Off	Does not display any object when not animating.
Visible When Off	Displays the specified object when not animating.
Current Frame When Off	Displays most recent animation frame when not animating.
First Frame When Off	Displays first frame of the animation when not animating.

Adding a Time/Date

To add time/date:

1. Select the object.
2. On the Dynamics menu, select Intrinsic > Time/Date.
3. Click in the work area. The Property Inspector appears with the Time/Date tab (Figure 11-14).

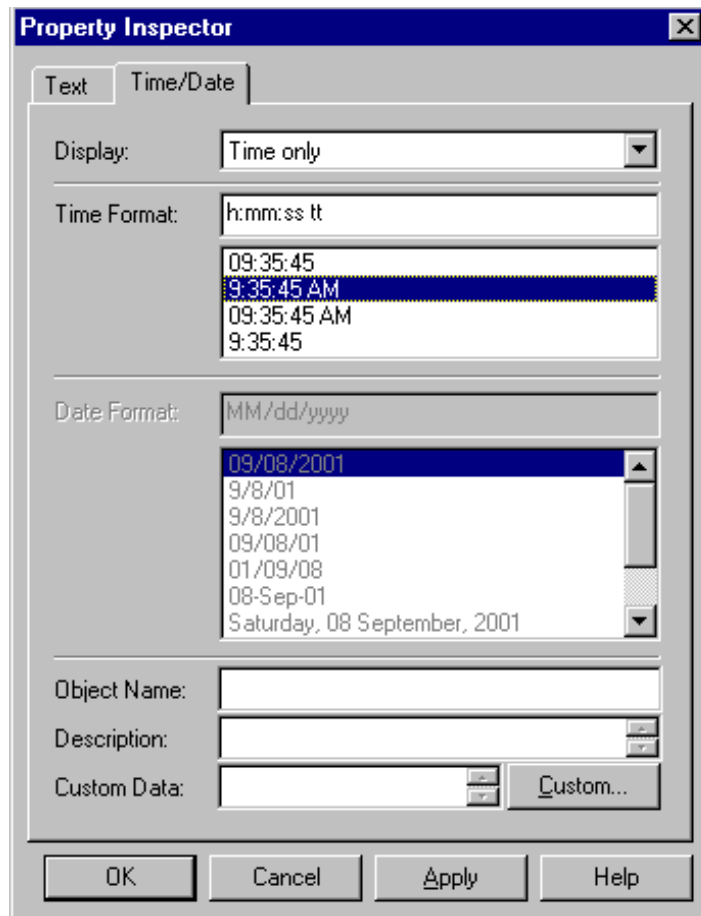


Figure 11-14: Property Inspector Dialog Box - Time/Date Tab

4. Select a time and date format.
5. Click OK.

Note: Additional properties relating to the visual appearance of the Time/Date object can be made by selecting the Text tab. The text page for Time/Date objects is similar to the page used to edit static text objects.

Creating a Process Point/Data Entry (PPT/DE)

To create a PPT/DE:

1. On the Dynamics menu, select Intrinsic > Process Point.
2. Click in the work area. The Property Inspector dialog box appears with the PPT/DE tab (Figure 11-15).

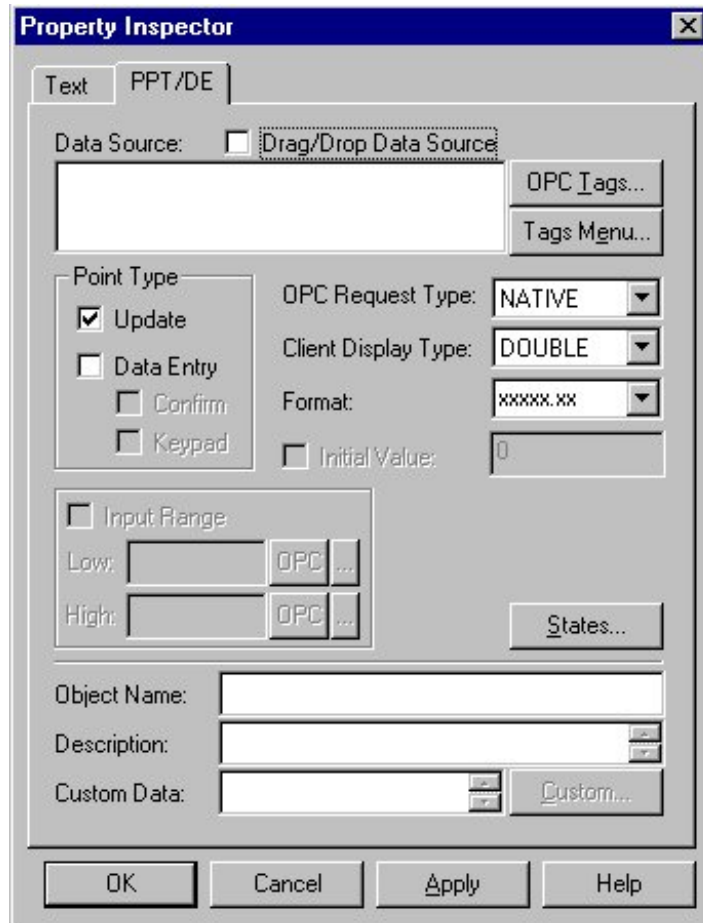


Figure 11-15: Property Inspector Dialog Box – PPT/DE Tab

3. Establish a data point connection.
4. Fill in the PPT/DE dynamic parameters using Table 11-15.
5. Click OK.

Table 11-15: Process Point/Data Entry Parameters

Parameter	Description
Data Source	Lists the Tag that drives this dynamic.
OPC Tags	Opens the OPC Universal Tag Browser to search for a Tag.
Drag/Drop Data Source	Specifies if the Data Source is eligible for drag and drop operation.
Tags Menu	Opens the Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Update	Defines a point type that the system updates realtime.
Data Entry	Allows the value of the connected point to be changed in Runtime mode.
OPC Request Type	Lists types of data requested by the OPC: float, double, Boolean, byte, word, dword, character, short, long, string, and native.
Client Display Type	Lists types of data displayed to the client: float, double, Boolean, byte, word, dword, character, short, long, and string.
Format	Type or select the value format for the connected point.
	Float* Lists 18 default formats such as xxxx.xx in a drop-down menu for analog (floating point) process points. Define a scientific notation format using the predefined formats, or create a user-defined format.
	Double* Lists the same formats as Float.
	Bool* Defines 2-state display formats. Offers default formats such as 0/1, Off/On, No/Yes, False/True, Auto/Manual, Disable/Enable, and Stop/Run.
	Byte* Lists ten default formats for integers such as xxxx.
	Word* Lists the same formats as Byte.
	Dword* Lists the same formats as Byte.
	Char* Lists the same formats as Byte.
	Short* Lists the same formats as Byte.
	Long* Lists the same formats as Byte.
	String* Defines String text display formats, 10, 20, 30, 40, 50, 60, 70, 80, as the number of characters. Define the number of text characters before you place this object in the display. Use the predefined format or create your own.
	Native** Defines native format for the data type.
Initial Value	Defines the initial (startup) value to which the data entry object is initialized when the display is launched in Runtime mode.
Input Range	Specifies a low and high range in the boxes provided. This range applies to data entries only. In Runtime mode, M-Graphics will not allow you to enter a value outside this range.
States	Opens the State Field Configuration dialog box to associate text with a numeric input value.

Notes: *Available data types for OPC Request Type and Client Display Type.

**Available for only the OPC Request Type.

Additional properties relating to the visual appearance of the PPT/DE can be made by selecting the Text tab. The text page for Process Points is similar to the page used to edit static text objects.

Creating a Pushbutton

To create a pushbutton:

1. On the Dynamics menu, select Intrinsic > Pushbutton.
2. Click in the work area. The Property Inspector appears with tabs for Pick (same tab as the pick dynamic) and Button (Figure 11-16).

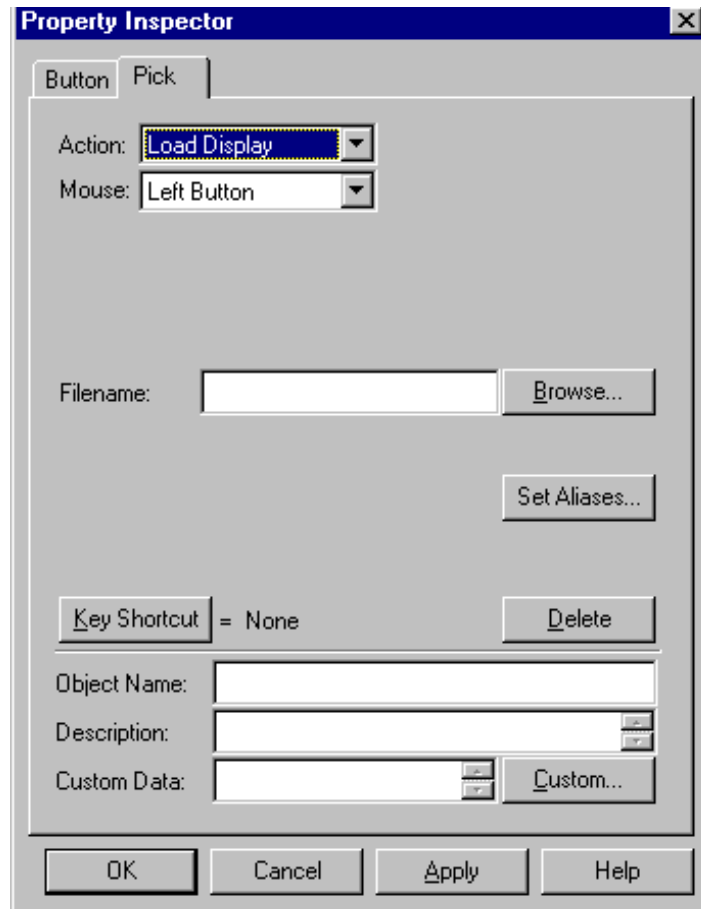


Figure 11-16: Property Inspector for Pushbutton Dynamic

3. Fill in the parameters.
4. Click OK.

Creating a Checkbox

To create a checkbox:

1. On the Dynamics menu, select Intrinsic > Checkbox.
2. Click in the work area. The Property Inspector appears with tabs for Pick (same tab as the pick dynamic) and Checkbox (Figure 11-17).

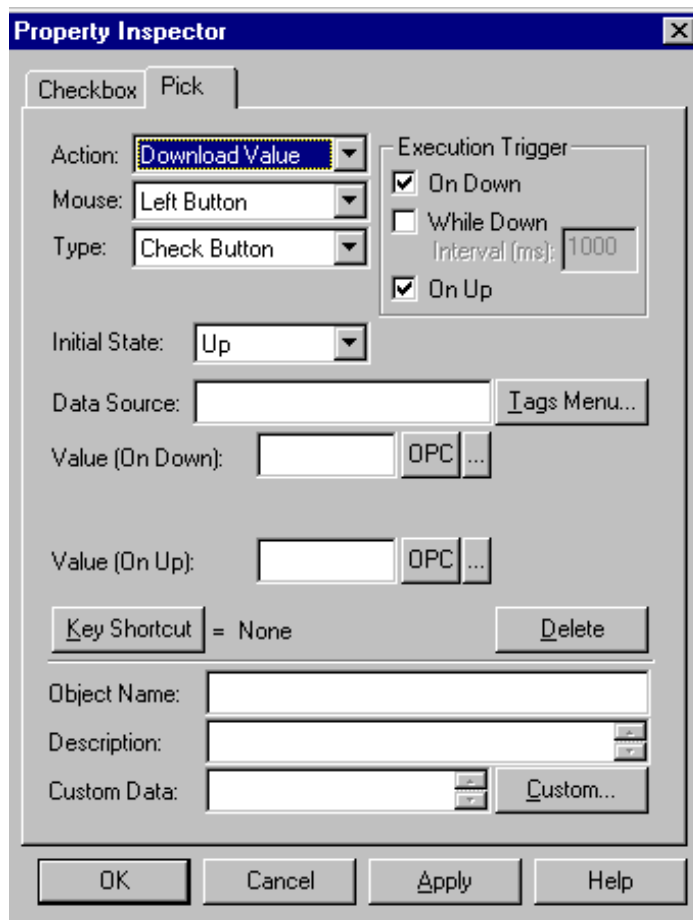


Figure 11-17: Property Inspector for Checkbox Dynamic

3. Fill in the parameters using Table 11-16.

Table 11-16: Pick Parameters

Field	Description
Action	Lists all the Pick action options: load display, drag/drop load, display back, display forward, pop-up window, embedded window, close window, launch application, download value, toggle value, set aliases, alias dialog, run VBA Script, and custom command.
Mouse	Lists the mouse button activating the pick (left, middle, or right).
Type	Lists all the Pick action options.
Execution Trigger	Displays when the value returns: on down, while down, or on up.
Initial State	Displays current state of action item.
Data Source	Displays the name of the current file.
Tags Menu	Opens the Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Value (On Down)	Displays the return value when down.
OPC	Opens the OPC Universal Tag Browser.
Value (On Up)	Displays the return value when up.
Key Shortcut	Defines a key shortcut for this action.
Delete	Deletes the dynamic.
Object Name	Identifies the object for OLE Automation.
Description	Describes the object and determines the ToolTips displayed.
Custom Data	Allows user to enter data.

4. Click OK.

Creating a Radio Button

To create a radio button:

1. On the Dynamics menu, select Intrinsic > Radio Button.
2. Click in the work area. The Property Inspector appears with tabs for Pick (same tab as the pick dynamic) and Radio Button (Figure 11-18).

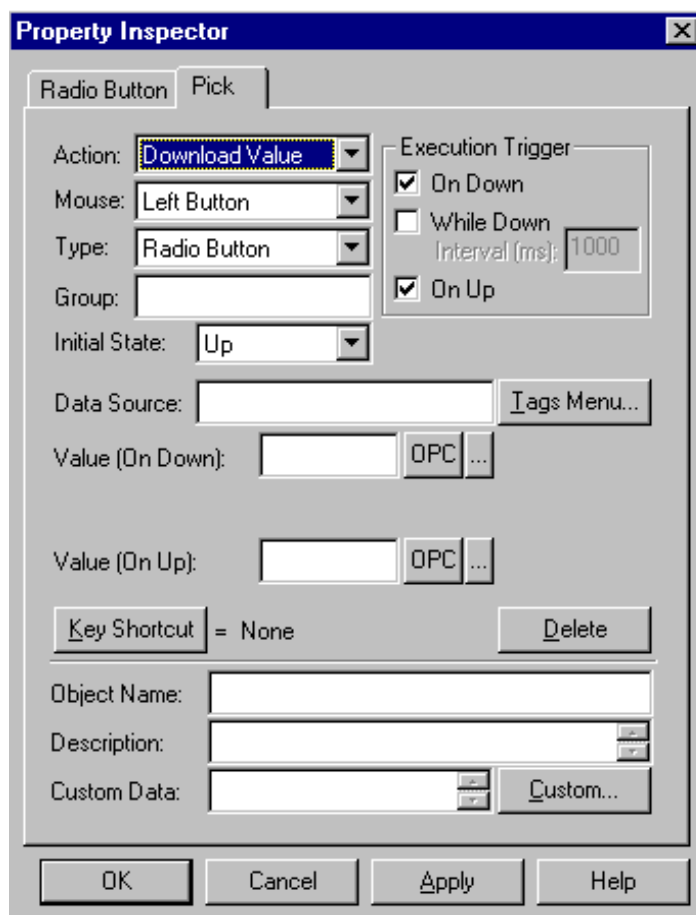


Figure 11-18: Property Inspector for Radio Button Dynamic

3. Fill in the parameters using Table 11-17.

Table 11-17: Radio Button Parameters

Field	Description
Action	Lists all the Pick action options: load display, drag/drop load, display back, display forward, pop-up window, embedded window, close window, launch application, download value, toggle value, set aliases, alias dialog, run VBA Script, and custom command.
Mouse	Lists the mouse button activating the pick (left, middle, or right).
Type	Lists all the Pick action options.
Group	Displays the group where the action returns.
Execution Trigger	Displays when the value returns: On Down, while down, or on up.
Initial State	Displays current state of action item.
Data Source	Displays the name of the current file.
Tags Menu	Opens the Expression Editor, Aliases, Local Variables, or Simulation Variables dialog box.
Value (On Down)	Displays the return value when down.
OPC	Opens the OPC Universal Tag Browser.
Value (On Up)	Displays the return value when up.
Key Shortcut	Defines a key shortcut for this action.
Delete	Deletes the dynamic.
Object Name	Identifies the object for OLE Automation.
Description	Describes the object and determines the ToolTips displayed.
Custom Data	Allows user to enter data.

4. Click OK. One of the radio buttons is always selected.

Creating a Display Button

To create a display button:

1. On the Dynamics menu, select Intrinsic > Display Button Wizard.
2. Click in the display. The Display Button Wizard appears (Figure 11-19).

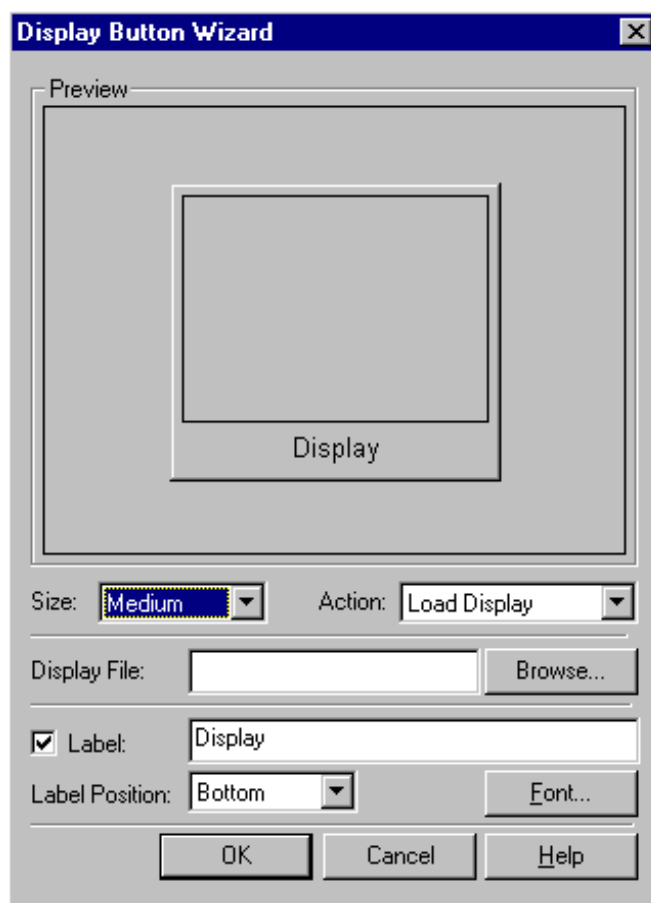


Figure 11-19: Display Button Wizard

3. Fill in the parameters using Table 11-18.
4. Click OK.

Table 11-18: Display Button Wizard Parameters

Parameter	Description
Size	Determines size of button: very small, small, medium, large, very large.
Action	Determines button's action: load display, embedded window, pop-up window, and drag/drop load.
Display File	Lists the file the display button loads.
Label	Determines if a label appears on the button and what that label says.
Label Position	Determines if label appears on the top or bottom.
Font	Determines font of the label.

Chapter 12

Using Visual Basic for Applications (VBA)

Introduction

Visual Basic for Applications (VBA) is a Microsoft® Visual Basic® programming system application. VBA is an industry standard and a powerful programming environment. It is the fastest and easiest way to create and customize Microsoft Windows® applications. M-Graphics is shipped with Microsoft VBA Release 5.0. VBA allows users to customize M-Graphics to suit their specific requirements. It also offers high-level application programmability and features cross-platform support for ActiveX technology®. It is identical with VBA in Microsoft Office applications and other third-party products.

This chapter describes how to:

- create and edit VBA scripts
- add event handlers
- display the *ThisDisplay* module code
- create a macro
- edit a macro
- step a macro
- run a macro
- clean VBA unused modules

Key Concepts

VBA Features

VBA allows you to:

- create, debug, and run custom scripts or macros
- write Visual Basic code for M-Graphics events
- modify M-Graphics native objects
- connect ActiveX objects to each other and to M-Graphics native objects

VBA allows both Configure and Runtime operations.

Windows

A window is a rectangular region with its own boundaries. Examples of windows include: a Windows Explorer window, a document window within Microsoft Word, on M-Graphics displays, a dialog box, text box, message box, command button, etc. A container is a window that contains one or more other windows, buttons, controls, etc. The Microsoft Windows operating system manages all the windows by assigning a unique ID to each window.

Events

Events are actions associated with a window. Events can occur through user actions such as mouse clicks, key presses, or as a result of another window's actions. When an event occurs, a message is sent to the operating system that broadcasts the message to other windows. Each window can take appropriate action based on its own instructions. Examples of event handling are: repainting of a window by itself when uncovered by another window, closing, minimizing or maximizing a window by clicking on the appropriate control, etc.

Many of the standard events or messages are handled automatically by VBA. Other events are exposed to the user as Event procedures, and the user can write powerful code to deal with Event procedures without having to deal with unnecessary details.

Object Oriented Programming

Visual Basic is an object oriented programming language. Contrary to procedural languages like C or BASIC, VB uses objects to create applications. Examples of objects are: forms, controls, M-Graphics displays, databases, etc. Create your own objects from a set of rules called classes. Objects and classes simplify your coding and increase code reuse.

Classes

Classes are sets of rules that define objects. Objects in Visual Basic are created from classes. An object is an instance of a class. The class defines an object's interfaces, default methods, and properties. Descriptions of classes are stored in type libraries and can be viewed with object browsers.

Objects

Objects are encapsulated, which means they contain both their code and their data. This makes them easier to maintain than traditional code. Visual Basic objects have properties, methods, and events.

Methods and Properties

Properties are data that describe an object. Methods are object actions. Events are actions the object performs.

Development Using Visual Basic for Applications

VBA uses an event driven model approach for development. Visual Basic interprets the code, as you write it. Write, compile, and test code during the development.

VBA Editor

VBA Editor is an Integrated Development Environment (IDE) and is integrated into M-Graphics. It can be opened from the M-Graphics menu bar.

Create, edit, debug, and run Visual Basic code using the Visual Basic Editor. The custom code is stored in modules, class modules, and forms.

The VBA Editor supports project management. Create projects using the Editor. Projects can contain M-Graphics objects, VB modules, forms, windows, etc. Projects can be opened or closed from the View menu.

VBA Editor Menu Bar

The menu bar displays the commands used with VBA. It consists of several menus: File, Edit, View, Window, Help, Project, Format, and Debug.

VBA Editor Context Menus

The context menus can be invoked by right-clicking an object. The context menus contain shortcuts to frequently performed actions.

VBA Editor Toolbars

The toolbars provide quick access to commands in the programming environment. By default, the Standard toolbar is displayed. Additional toolbars for editing, form design, and debugging toggles on or off the Toolbars command on the View menu.

VBA Editor Toolbox

The toolbox provides a set of tools used at design time to place controls on a form. In addition to the default toolbox layout, create custom layouts by selecting Add Tab from the context menu and adding controls to the resulting tab.

VBA Editor Project Explorer Window

The Project Explorer Window lists the forms and modules in the current project. As you create, add, or remove files from a project, Visual Basic reflects the changes in the Project Explorer window, which contains a current list of the files in the project.

Project

A project is the collection of files used to build an application. In the VBA project, there is a collection of modules; all modules are stored, along with M-Graphics objects, in the same file (.gdf).

However, it is possible to export the modules to .bas files, the class modules to .cls files, and forms to .frm files. A project consists of:

- one project file that keeps track of all the components (.vbp).
- one file for each form (.frm).
- one binary data file for each form containing data for properties of controls on the form (.frx). These files are not editable and are automatically generated for any .frm file that contains binary properties such as Picture or Icon.
- optionally, one file for each class module (.cls).
- optionally, one file for each standard module (.bas).
- optionally, one or more files containing ActiveX controls (.ocx).
- optionally, a single resource file (.res).

Project File

The project file is a list of all the files and objects associated with the project as well as information on the environment options. This information is updated every time you save the project. All of the files and objects can be shared by other projects.

Properties Window

The Properties window lists the property settings for the selected form or control. A property is a characteristic of an object, such as size, caption, or color.

Object Browser

The Object Browser lists objects available for use in a project and provides quick navigation through your code. Use the Object Browser to explore objects in Visual Basic and other applications, view methods and properties are available for those objects, and paste code procedures into the application.

Form Designer

The form designer serves as a window that you customize to design the interface of your application. Add controls, graphics, and pictures to a form to create the look you want. Each form in an application has its own form designer window.

Code Editor Window

The code editor window serves as an editor for entering application code. A separate code editor window is created for each form or code module in an application.

Form Layout Window

The form layout window positions the forms in an application using a small, graphical representation of the screen.

Immediate, Locals, and Watch Windows

These additional windows are provided for use in debugging an application. They are only available when running an application within the editor.

Forms and Controls

Forms are user interfaces, which are the visual part with which the user interacts. Forms and controls are the basic building blocks used to create the interface.

Forms are objects that expose properties defining their appearance, methods defining their behavior, and events defining their interaction with the user. By setting the properties of the form and writing Visual Basic code to respond to its events, you customize the object to meet your requirements.

Controls are objects contained within form objects. Each type of control has its own set of properties, methods, and events that make it suitable for a particular purpose. Examples of controls are fields for entering or displaying text. Controls can also be used for accessing other applications and process data as if the remote application was part of your code.

ActiveX

ActiveX is a set of integration technologies that enables software components to interoperate in a networked environment using any language. ActiveX is based on Microsoft Object Linking and Embedding (OLE) and the Component Object Model (COM).

ActiveX Control

ActiveX is a type of control and is an extension to the Visual Basic Toolbox. When adding an ActiveX control to a program, it becomes part of the development and Runtime environment and provides new functionality for your application.

ActiveX Used with M-Graphics

ActiveX control is used to embed documents from other applications into M-Graphics displays. Applications supporting ActiveX include Iconics TWXView32 or other Windows applications.

Conversely you could open and run M-Graphics displays from other applications such as Microsoft Internet Explorer.

Modules

Code in VBA is stored in modules. There are three kinds of modules: form, standard, and class. By default, the VBA modules are stored in .gdf files. They can be exported to files and imported back.

Standard Module

Usually the code associated with a form resides in that form module. If you have forms or other modules that could use a common code, create a separate module containing a procedure that implements the common code. This separate module should be a standard module.

Each standard module can contain declarations such as, type, variable and procedures such as Function (functions) or Sub (subroutines). The standard module file has an extension .bas.

Form Modules

Form modules are the foundation of most Visual Basic applications. They can contain procedures that handle events, general procedures, and form level declarations of variables, constants, types, and external procedures. The code in a form module is specific to the particular application to which the form belongs; it might also reference other forms or objects within that application. The form module files have an extension .frm.

Class Modules

Class modules (.cls file name extension) are the foundation of object-oriented programming in Visual Basic. Write code in class modules to create new objects. These new objects can include customized properties and methods. Forms are just class modules that can have controls placed on them and can display form windows.

ActiveX Modules

Various ActiveX modules include ActiveX Documents, ActiveX Designers, and User Controls. From the standpoint of writing code, these modules should be considered the same as form modules.

M-Graphics VBA Project

The M-Graphics VBA project is loaded whenever you launch the VBA Editor from M-Graphics. This project contains groups of modules by default such as:

- M-Graphics Native Objects
- Modules - ThisDisplay module and GwxTools module
- Forms

Each module can contain VBA code – functions, subroutines, event handlers, and global declarations.

GwxTools Module

GwxTools is a custom module with common subroutines used in the VBA Wizard described in the next chapter. The GwxTools module is not available to the user until a VBA Wizard is used.

ThisDisplay Module

ThisDisplay represents the current M-Graphics display.

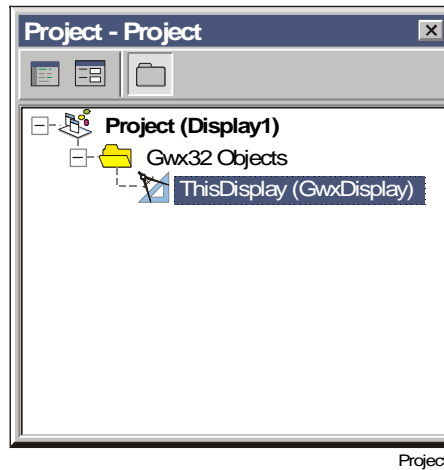


Figure 12-1: Project Window with *ThisDisplay* Module

The VBA programmer has full control over the properties and native objects of the current display and can control the rich animation interface of M-Graphics.

The Automation properties and methods are accessible through the *ThisDisplay* module. *ThisDisplay* is a representation of a current M-Graphics display and contains all Automation properties and methods.

VBA can display Automation methods and properties only if the Automation object is checked in the References dialog box.

Note M-Graphics is checked automatically for every new display. Refer to Figure 12-2.

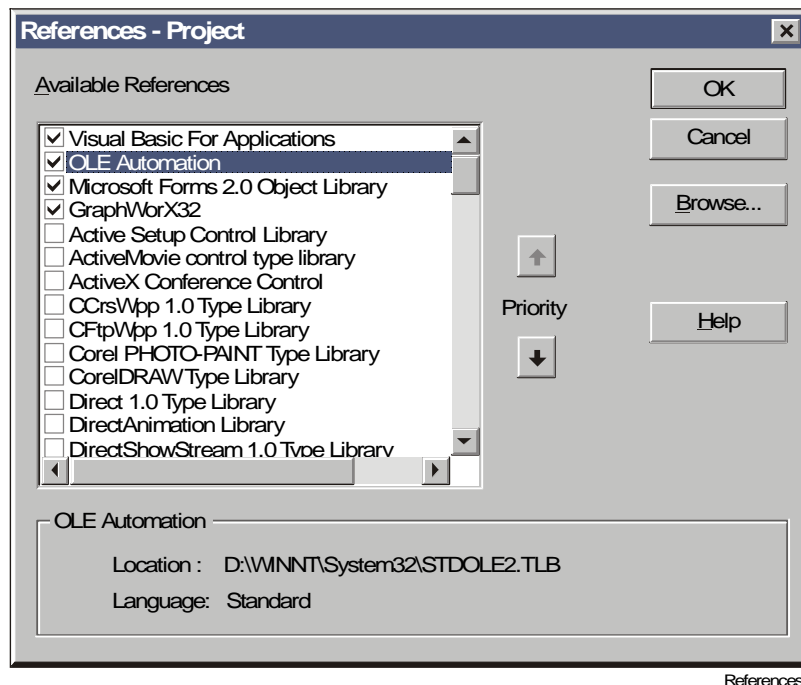


Figure 12-2: References Dialog Box

M-Graphics Native Display Objects

M-Graphics native objects like ellipses, rectangles, symbols, and dynamic actions are not exposed to VBA by default. However they can be referenced from VBA if they have an *Object Name* assigned through Property Inspector dialog box.

A reference (actually a dispatch pointer) to a named object can then be retrieved by one of following methods of the *ThisDisplay*, *Symbol*, and *Visible* objects (Table 12-1).

Refer to the *OLE Automation Reference* section of this manual for more information on these functions.

Table 12-1: ThisDisplay, Symbol, and Visible Methods

Method	Description	Example
ThisDisplay.GetVisibleObjectFromName("Name").	This function takes the name of a visual object in the display and reports back the object associated with that name. Upon storing the result of this function in a variable, can change the properties of the object in the display by changing the fields of the variable. The following example rotates a rectangle named "Square" 45 degrees from horizontal.	<pre>Dim obj As GwxRectangle Set obj = ThisDisplay.GetVisibleObjectFromName("Square") obj.Angle = 45</pre>
ThisDisplay.GetDynamicObjectFromName("Name").	This function takes the name of a dynamic object in the display and reports back the dynamic object associated with that name. A dynamic object is not a physical, observable object but an abstraction for the operation performed on a visible object (i.e., Hide, Rotation, Size). Once the dynamic object is stored in a variable, you can change its properties. The following example changes the data source of an M-Graphics Hide object to zero hiding the visible object associated with the dynamic object.	<pre>Dim obj As GwxHide Set obj= ThisDisplay.GetDynamicObjectFromName("hd") obj.dataSource = 0</pre>
ThisDisplay.GetVisibleObjectFromIndex(Long Value).	This function selects an object based on the order in which visible objects on the screen were created. (The first visible object put on the screen has an index of 0.) This function is useful for iterating through all the objects in a display. The following code turns the first object created to green and the second to red.	<pre>Dim obj1 As Object Dim obj2 As Object Set obj1 = ThisDisplay.GetVisibleObjectFromIndex(0) Set obj2 = ThisDisplay.GetVisibleObjectFromIndex(1) obj1.fillColor = RGB(0, 255, 0) obj2.fillColor = RGB(255, 0, 0)</pre>
Symbolname.GetVisibleObjectFromIndex(Long value)	Each M-Graphics symbol has its own index that keeps track of the objects within it. The GetVisibleObjectFromIndex method, when appended to the name of a symbol, finds the visible object within the symbol with the specified index. This function is useful for iterating through all objects in a symbol. The following example turns the third visible object in a symbol named "sym" to green.	<pre>Dim sym1 As GwxSymbol Dim obj As Object Set sym1 = GetVisibleObjectFromName("sym") Set obj = sym1.GetVisibleObjectFromIndex(2) obj.fillColor = RGB(0,255,0) (Note that the third object has index of 2. First has index 0.)</pre>
Symbolname.GetVisibleObjectFromName("Name")	Finds a visible object within a symbol by the object name specified in M-Graphics.	
VisibleObjectName.GetDynamicObjectFromIndex(Long value)	Every time a dynamic object is assigned to a unique visible object, it is assigned an index. The first dynamic object assigned is given an index of zero. Since one visible object can be associated with many dynamic objects, this function provides a useful way of manipulating dynamic objects. The following code takes the second dynamic object associated with a rectangle named "rect" and changes its low range to 10.	<pre>Dim o_Vis As GwxRectangle Dim o_Dyn As Object Set o_Vis = ThisDisplay.GetVisibleObjectFromName("rect") Set o_Dyn = o_Vis.GetDynamicObjectFromIndex(1) o_dyn.lowRange = 10</pre>
VisibleObjectName.GetDynamicObjectFromName("Name")	This function takes the name of a dynamic object associated with a visible object and allows you to represent the dynamic object with a variable.	

Table 12-2: Additional VBA Events for Layering

Event	Description
PreAnimateLayer(BSTR layerName)	Fired before data for the layer is requested.
PostAnimateLayer(BSTR layerName)	Fired after data for the layer has been requested.
PreDeanimateLayer(BSTR layerName)	Fired before data for the layer is released.
PostDeanimateLayer(BSTR layerName)	Fired after data for the layer has been requested.

Note: For more information on Layers, refer to the *Arranging Objects* chapter.

Unique Object Names

Sometimes it is desired to access the same objects in all duplicates (clones) of the desired symbol in the same way. This technique is useful for VBA Wizards that have a macro behind symbol. This macro is shared by all duplicates of the symbol and can be run on any of these duplicates.

To allow this feature, M-Graphics supports partial names for objects in symbols. The partial name is a name that ends with an underscore (e.g., MyEllipse_). The duplicates of MyEllipse_ are then MyEllipse_1, MyEllipse_2, What is important is the symbol method `GetVisibleObjectFromSymbol(partialName)` which accepts this partial name and returns the first occurrence of the specified object in specified symbol.

Additional Information

For VBA programming, working with the modules and forms, and customizing of the VBA Editor, refer to *Microsoft Visual Basic 5.0 Programmer's Guide* or documentation on VBA and the help file that comes with VBA. You can open it from the VBA Editor from the menu bar. You can also refer to any VB and VBA document containing useful information.

Note that the VBA in M-Graphics is the same as the one in Microsoft Office applications (Word, PowerPoint®, Access, Excel) and other products. Once you master VBA in M-Graphics, you are able to program in all other applications.

There are many examples included with M-Graphics that are good sources of information and VBA programming tips and tricks.

It is also possible and helpful to open two or more instances of M-Graphics; in each instance open the VBA Editor, and copy and paste the VBA code between the instances.

VBA Wizards

VBA Wizards are M-Graphics objects with Visual Basic code behind them. The code is run either in Design mode to help to configure M-Graphics object or in Runtime mode to execute a specific task.

Normally, the VBA code is stored in the current document, saved from the VBA Editor when the display is saved, and re-loaded to VBA Editor when a display is open in M-Graphics.

However, if the rules described below are followed, the VBA code can be bound to an M-Graphics object. When such an object is pasted/dropped to another instance of M-Graphics, to Symbol Library, or to scrap (desktop), this code goes with it.

The VBA Wizard can be run to perform a specific task either in Design or Runtime mode.

Design Mode

M-Graphics can launch the VBA Wizard macro in Design mode by double-clicking the VBA Wizard.

By default when an object is double-clicked in Design mode, a Property Inspector is launched. However, if there is a special keyword in the first line of the Custom Data field of Property Inspector, a macro can be run.

The format of the key word for a macro *MacroName* is:

```
OnDoubleClick=<GwxMacroName_Main.MacroName>, Parameters=<>
```

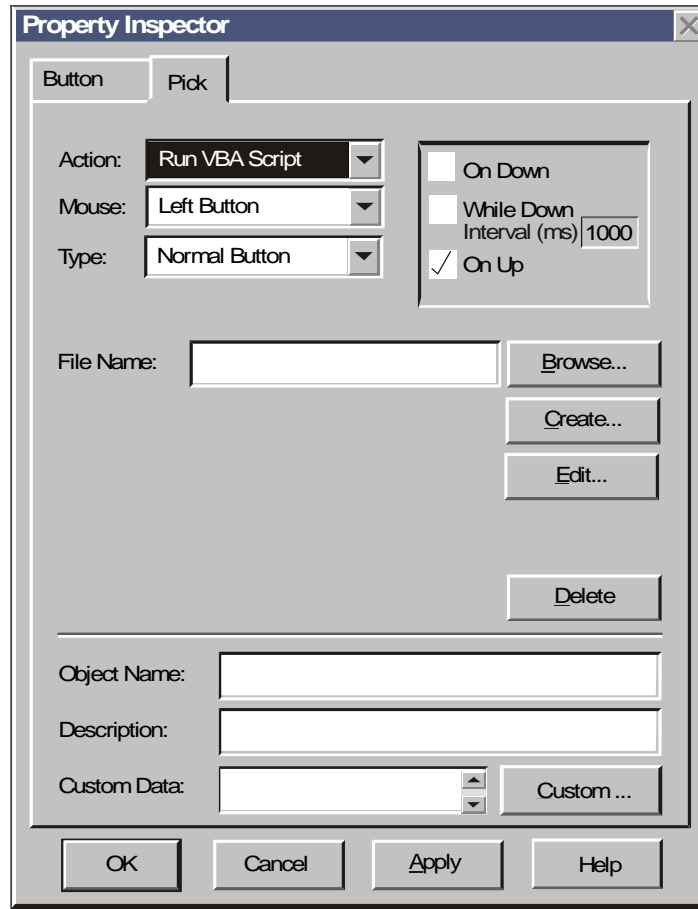
There must exist a macro *MacroName* in the module *GwxMacroName_Main* in VBA to successfully run the macro. The user is allowed to put any string between angle brackets of a *Parameters=<>* section. Any data worth sending to the macro would be put between the angle brackets. These data are then available when the macro runs.

The name of a macro cannot contain spaces.

Runtime Mode

M-Graphics can launch the VBA Wizard macro in Runtime mode by clicking an M-Graphics button or a pick action configured to run a macro. The Action field must be configured to Run VBA Macro. The Macro Name field must contain a macro name in its format:

```
GwxMacroName_Main.MacroName
```



Prop vba

Figure 12-3: VBA Wizard in Button

Custom Data can contain any string keeping custom data as desired between angle brackets.

VBA Wizard Rules

M-Graphics takes care of code behind the VBA Wizard. Due to the special naming convention of all modules belonging to a specific VBA Wizard, the code in these modules can be properly moved with the symbol.

If the macro name of the VBA Wizard is *MacroName*, then all code modules must start with the string *GwxMacroName_*. This technique allows more code and form modules to be used for one VBA Wizard object and facilitates moving all this code with the object when necessary.

VBA Wizard Creation Tool - Macro

Because creating VBA Wizard objects would be a tedious task, M-Graphics offers a useful creation tool: a macro that converts the object to VBA Wizard and generates the VBA template code. The code can then be easily enhanced and modified by the user.

When a VBA Wizard is deleted or moved out of the current display, the VBA code is not deleted automatically. However, you can use Clean Unused VBA Modules to remove all modules from the VBA Editor which start with the *Gwx* string and are not referenced from the currently displayed VBA Wizards (either Design or Runtime mode based).

If you want to create a new Runtime based VBA Wizard, add a pick dynamic with the Run VBA Script. Refer to the *Adding Dynamics* chapter of this manual.

Procedure Overview

Table 12-3: Using VBA Scripts

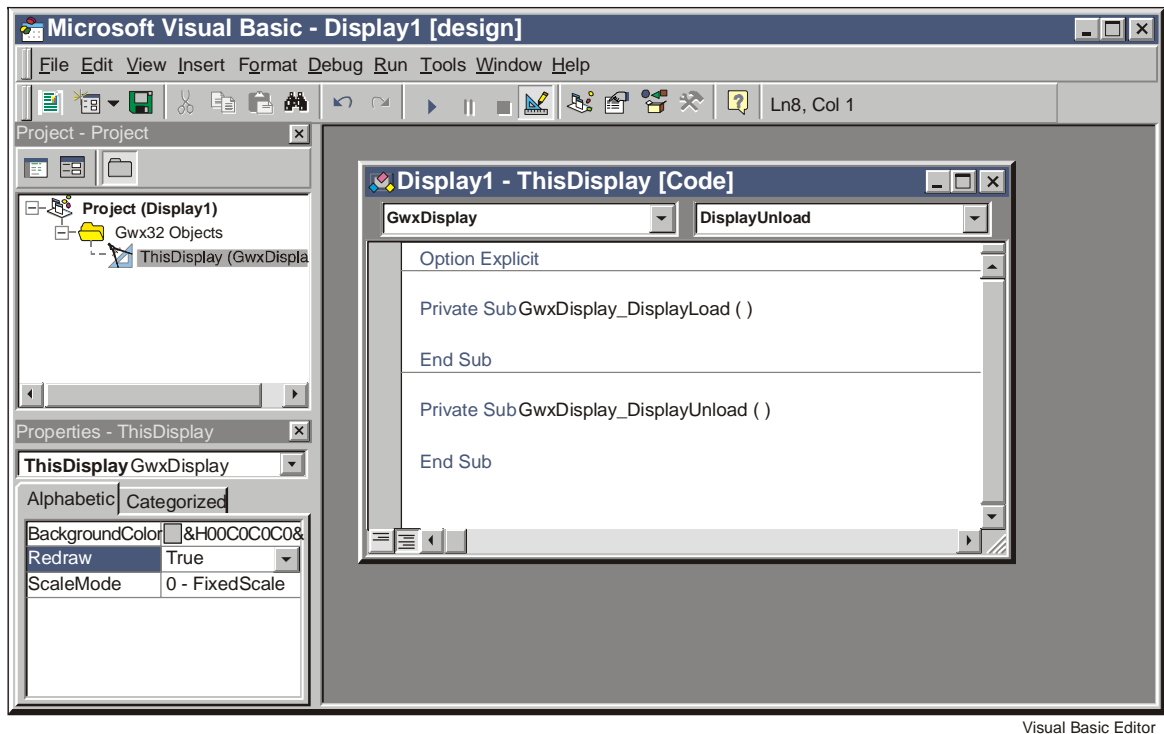
To Do This	Follow These Steps:
Create and Edit VBA Scripts	On the Tools menu, select Macros > Visual Basic Editor. Add code, forms, etc., to accomplish desired task.
Add Event Handlers	Open a display. On the Tools menu, select Macros > Visual Basic Editor. Open the <i>ThisDisplay</i> module from the VBA Editor. Select the <i>GwxDisplay</i> module in left combo box of <i>ThisDisplay</i> module from the VBA Editor. Select the desired event from the list in the right combo box. On the File menu, select Save Display 1. On the File menu, select Close and Return to M-Graphics.
Display the <i>ThisDisplay</i> Module Code	On the View menu of the VBA Editor, select Project Explorer. Expand the M-Graphics Objects group and double-click the <i>ThisDisplay (GwxDisplay)</i> item.
Create a Macro	Create a symbol. On the Tools menu, select Macros > Create Macros. Fill in the parameters. Click OK. Type code. Return to M-Graphics and double-click on symbol.
Edit a Macro	Select a macro object in the display. On the Tools menu, select Macros > Edit Macro.
Step a Macro	Select a macro object in the display. On the Tools menu, select Macros > Step Macro.
Run a Macro	Select a macro object in the display. On the Tools menu, select Macros > Run Macro.
Clean VBA Unused Modules	On the Tools menu, select Macros > Clean VBA Unused Modules.

Detailed Procedures

Creating and Editing VBA Scripts

To create and edit VBA Scripts:

1. On the Tools menu, select Macros > Visual Basic Editor. The Visual Basic Editor appears (Figure 12-4).



Visual Basic Editor

Figure 12-4: Visual Basic Editor

2. Add code, forms, etc., to accomplish desired task.

Adding Event Handlers

To add event handlers:

1. Open a display.
2. On the Tools menu, select Macros > Visual Basic Editor. The M-Graphics VBA Project loads by default.
3. Open the *ThisDisplay* module from the VBA Editor.
4. Select the *GwxDisplay* module in left combo box of *ThisDisplay* module.
5. Select the desired event from the list in the right combo box.

Note: The subroutine header is automatically inserted to current module. Insert your VBA code to the body of the subroutine.

6. On the File menu, select Save Display 1.
7. On the File menu, select Close and Return to M-Graphics.

Displaying the *ThisDisplay* Module Code

To display the *ThisDisplay* module code:

1. On the View menu of the VBA Editor, select Project Explorer.
2. Expand the M-Graphics Objects group and double-click the *ThisDisplay* (*GwxDisplay*) item. A *ThisDisplay* code window appears (Figure 12-5).

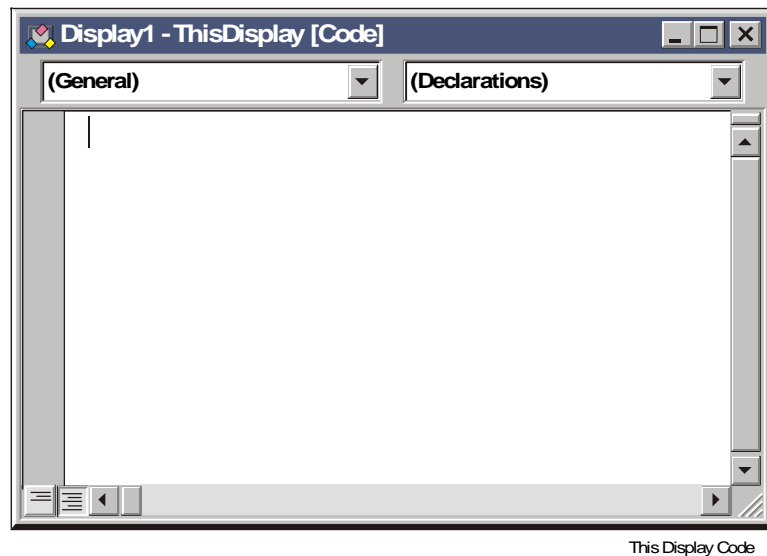


Figure 12-5: *ThisDisplay* Code Window

ThisDisplay contains two combo boxes at the top.

The top left combo box allows select items like:

(General)

GwxDisplay

The top right combo box shows events for the left combo selection. For example, the *GwxDisplay* item has events like *DisplayLoad*, *DisplayUnload*, and others.

Creating a Macro

To create a macro:

1. Create a symbol.
2. On the Tools menu, select Macros > Create Macros. The VBA Script Wizard dialog box appears (Figure 12-6).

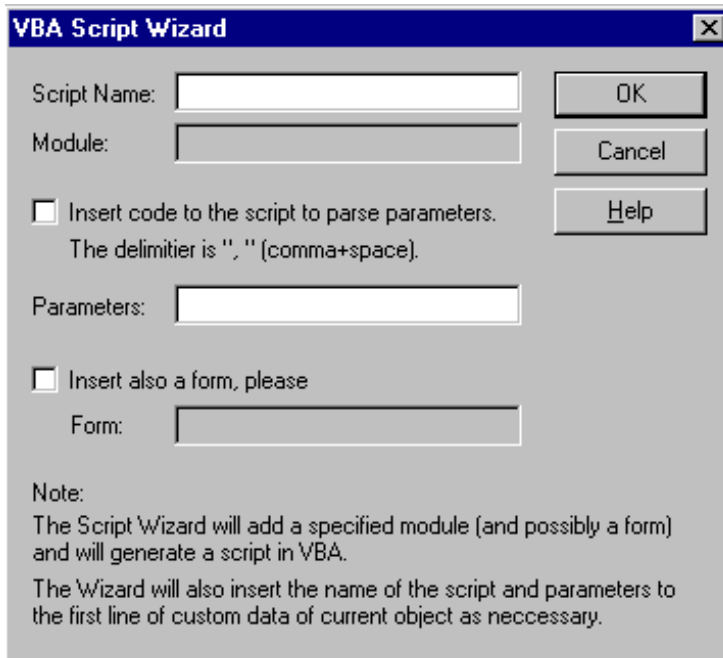


Figure 12-6: VBA Script Wizard Dialog Box

3. Fill in the parameters using Table 12-4.

Table 12-4: VBA Script Wizard Dialog Box Parameters

Parameter	Description
Script Name	Type the name of your script. The script name should begin with a character and should contain alphanumeric characters only. If there is already a macro or a module of that name, choose another name.
Module	Field Module is always grayed-out, because the module name is generated automatically based on macro name.
Insert code to the script to parse parameters.	Check this box to generate extra code in the body of the script subroutine. It helps in retrieving and storing parameters from the VBA Wizard object. This code uses the <i>GwxTools</i> module to convert parameters to a string named <i>StrPar</i> which is local to your script subroutine.
Parameters	Type any string you like. You can obtain this string when the macro runs. This field is designed to allow custom data specific to a VBA Wizard instance. Different instances of the same objects can keep different data. Using this field is optional. This feature simplifies your code by allowing you to pass values into a macro.
Insert also a form, please	Check this checkbox if you need a VBA form to be launched from the macro. You are allowed to create any number of forms for the VBA Wizard assuming you follow the naming convention. (If you don't follow the convention, the VBA code is not moved with the object when necessary.)
Form	Field Form is always grayed-out and generated automatically based on macro name.

4. Click OK. A VBA Editor starts, and the cursor appears in the body of *Test* subroutine in a module *GwxTest_Main*.
5. Type the code which is run when you double-click on the symbol in Design mode. Try the example code as shown in Figure 12-7.

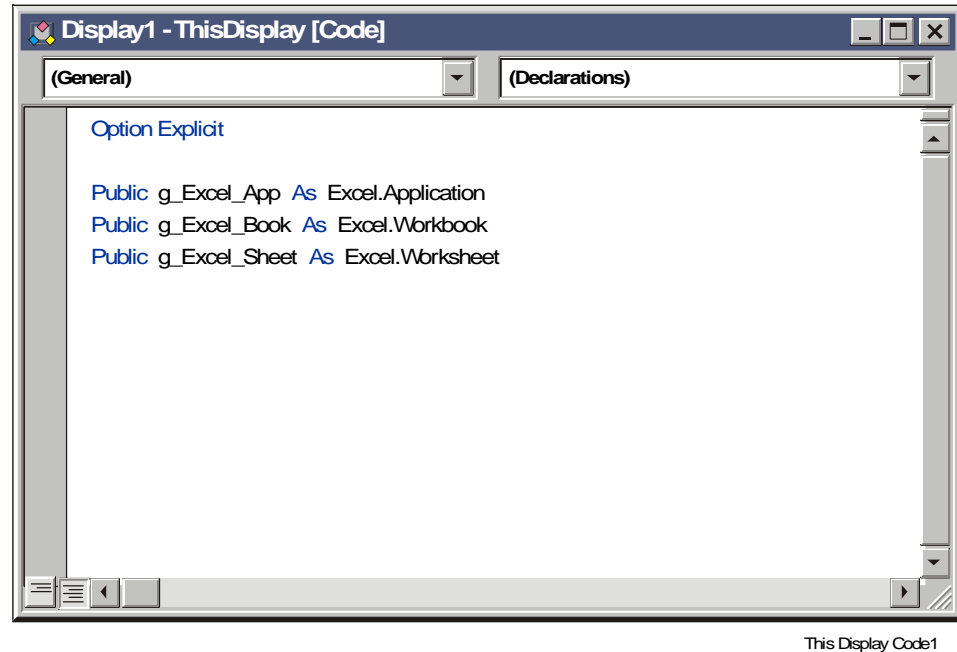


Figure 12-7: Test Subroutine

6. Return to M-Graphics, and double-click the symbol. A beep sounds, and a message appears.

Editing a Macro

To edit a macro:

1. Select a macro object in the display.
2. On the Tools menu, select Macros > Edit Macro. The VBA Editor appears with the cursor in the body of the macro.

Stepping a Macro

To step a macro:

1. Select a macro object in the display.
2. On the Tools menu, select Macros > Step Macro. The VBA Editor appears with the cursor on the first line of the macro.

Running a Macro

To run a macro:

1. Select a macro object in the display.
2. On the Tools menu, select Macros > Run Macro.

Cleaning VBA Unused Modules

To clean VBA unused modules:

On the Tools menu, select Macros > Clean VBA Unused Modules.

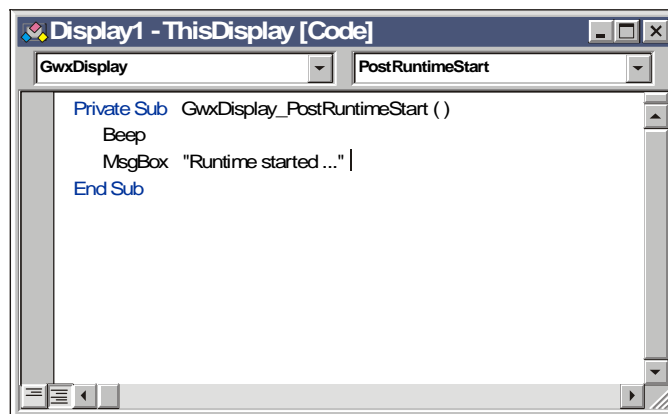
Examples

The following section provides several examples for reference.

Adding Event Handlers Example

Write code to pop-up a message box that displays the message `Runtime started` when you Runtime your display.

1. Open a display.
2. On the Tools menu, select Macros > Visual Basic Editor. The M-Graphics VBA Project loads by default.
3. Open the *GwxDisplay* module from the VBA Editor.
4. Select the *PostRuntimeStart* event from the list in the right combo box.
5. Insert VBA code as shown in the figure below to the body of the subroutine in the subroutine header (Figure 12-8).



Post Runtime Start

Figure 12-8: PostRuntimeStart Event Example

6. Switch to M-Graphics by closing the VBA Editor.
7. Test the event by clicking on Runtime on the menu bar.
8. You hear a beep, and a message box with a message Runtime started appears.

M-Graphics Native Display Objects Example

Write code for changing the color of ellipse M-Graphics native object during Runtime:

1. Load an existing display, or create a new M-Graphics display.
2. Draw an ellipse by picking Ellipse from the Draw toolbar.
3. Name the Ellipse Object for this display:
 - a. Double-click on the Ellipse. The Property Inspector box opens.
 - b. Type the name MyEllipse in the object name field and then click on OK.
4. Open VBA Editor.
5. Select the *ThisDisplay* module.
6. Select the *GwxDisplay* module in left combo box of *ThisDisplay* module.
7. Select the *PostRuntimeStart* event from the list in the right combo box.

8. The subroutine header is automatically inserted into the current module. Insert the VBA code as in the diagram below to the body of the subroutine.

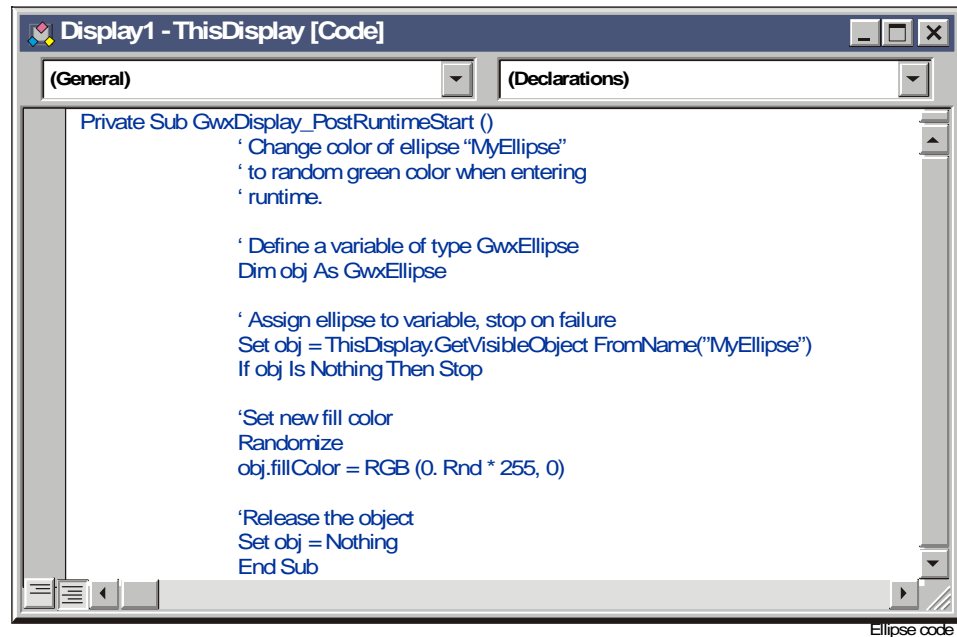


Figure 12-9: M-Graphics Ellipse Modified From VBA

9. Save the M-Graphics Project by selecting File - Save Display 1.
10. Test the example by switching to the display and going into Runtime. The color of the ellipse changes to green.

Unique Object Names Example

Assume you have a symbol which consists of a rectangle and an ellipse. Write code that modifies these objects in any copy of the symbol. (You must assign partial names to both objects, which name them MyRect_ and MyEll_.) Here is the code that shows how to access these objects in a specific symbol:

```

Dim sym As GwxSymbol
Set sym = FindSomehowDesiredSymbol()
' user method to choose the symbol
Dim ell As GwxEllipse, rect As GwxRectangle
Set ell= sym.GetVisibleObjectFromName("MyRect_")
Set rect= sym.GetVisibleObjectFromName("MyEll_")
' do something with these objects
' release references
Set ell = nothing
Set rect = nothing
Set sym = nothing

```

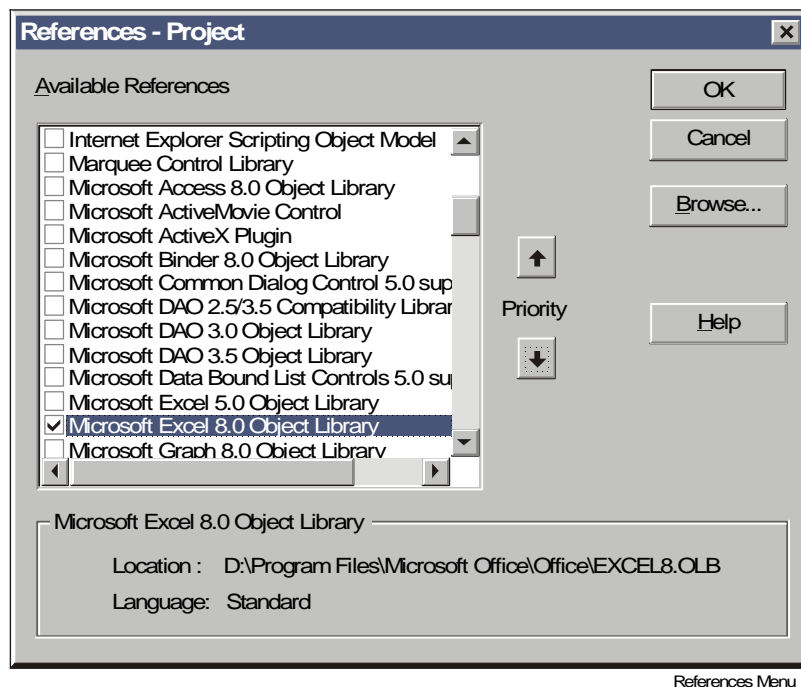
The following list demonstrates the unique name technique:

1. Create one rectangle, and assign object name Rect_1.
2. Duplicate this to create one more rectangle.
3. Group Rect_1 and Rect_2.
4. Duplicate this group using the Duplicate button.
5. Ungroup the duplicate object.
6. Read the Object name by using the Property Inspector dialog box. Notice that the Object names of the two rectangles (the third and the fourth) are Rect_3 and Rect_4.

Using VBA to Connect with Other Applications Example

Design a display and a spreadsheet, each with two data points, and have them communicate to each other through VBA.

1. Open a new M-Graphics project.
2. On the Tools menu, select Macros > Visual Basic Editor.
3. On the Tools menu, select References. This opens a list of available references to applications. Check the box next to Microsoft Excel Release 8.0 Object Library.

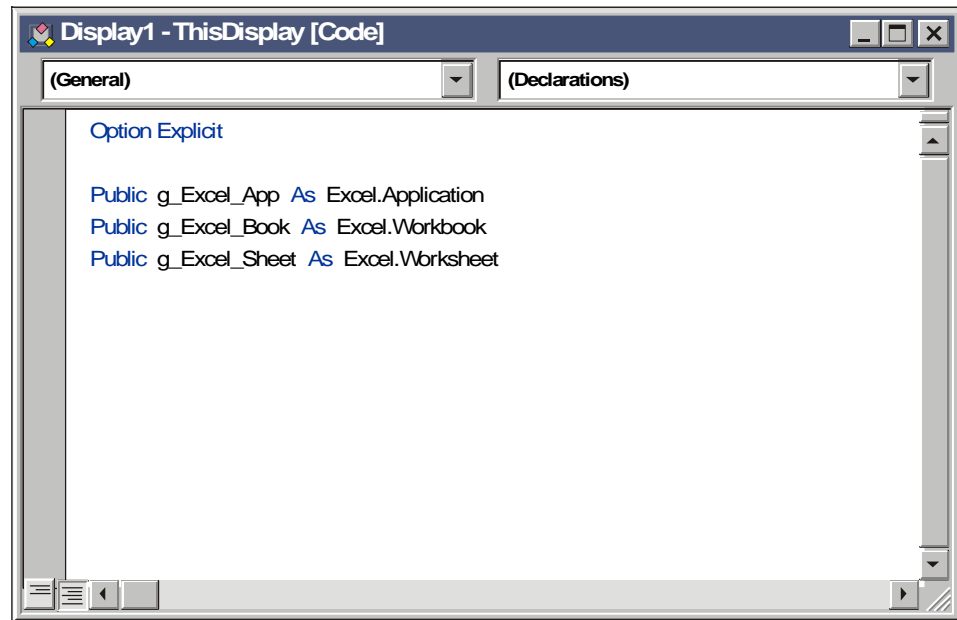


References Menu

Figure 12-10: References Menu

4. In the Project viewer window, double-click on the *ThisDisplay* Module. You should see a code window with the words `Option Explicit`. In that window, you want to make global declarations that will be used later:

```
Option Explicit
Public g_Excel_App As Excel.Application
Public g_Excel_Book As Excel.Workbook
Public g_Excel_Sheet As Excel.Worksheet
```



This Display Code1

Figure 12-11: ThisDisplay Code Window with Global Declarations

5. In the left combo box at the top of the code window, select `GwxDisplay`. In the right combo box, select `PreRuntimeStart`. By entering the following code, you cause Excel to be launched before M-Graphics goes into Runtime.

```
Private Sub GwxDisplay_PreRuntimeStart()
    ' Open up Excel and make it visible
    Set g_Excel_App =
CreateObject("Excel.application")
    g_Excel_App.Visible = True
    ' Open up a sheet
    Set g_Excel_Book = g_Excel_App.Workbooks.Add
    Set g_Excel_Sheet = g_Excel_Book.Worksheets(1)
    ' Initialize the two cells you will be using
    g_Excel_Sheet.Range("a1") = 0
    g_Excel_Sheet.Range("a2") = 0
End Sub
```

6. Create two process points in your display. Make sure that the Data Entry checkbox is checked for both process points. Connect one point to a local variable `~~a1~~` and the other to a local variable named `~~a2~~`. This double tilde is a standard notation for local variables in M-Graphics.

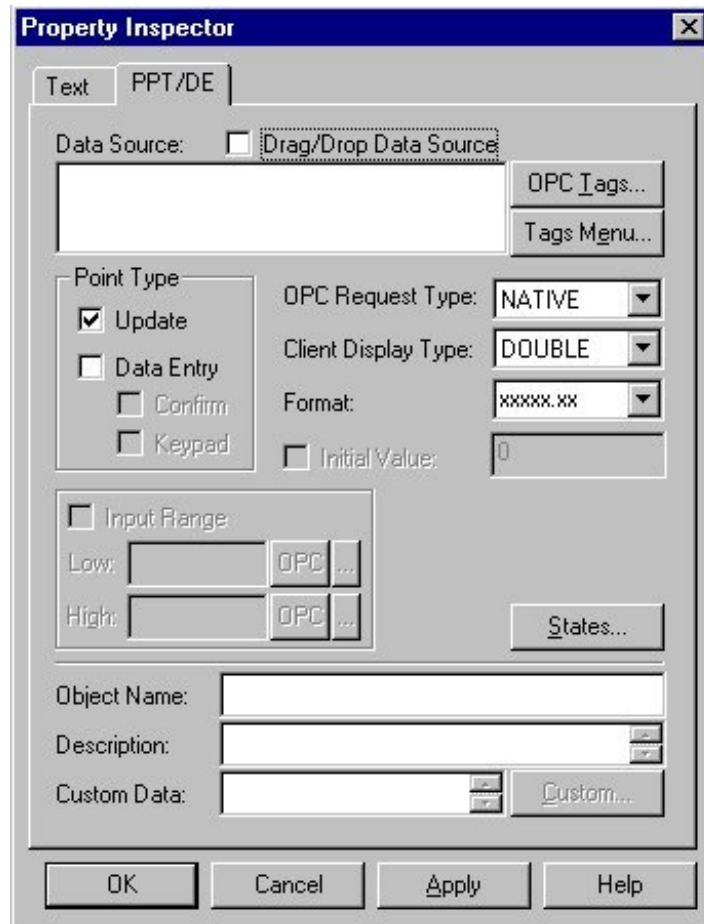


Figure 12-12: Configuration of Process Point

7. Now go back to the Visual Basic editor.
8. Make a new procedure designed to read from and write to Excel. Go to Insert-Procedure in the menu bar.

9. Name the procedure `Read_Write`, and enter in the following code:

```
Public Sub Read_Write(ByVal Co As Integer)
    ' This procedure reads from or writes to excel
    ' based on the condition, Co, passed as a
    ' parameter

    Dim Point As GwxPoint
    Dim St As String 'Used to store cell name
    Dim St2 As String 'Used to store variable name
    Dim X As Integer
    For X = 1 To 2
        ' add "a" to the value X converted to string
        St = "a" + Mid(Str(X), 2, 1)
        St2 = "~~" + St + "~~"
        Set Point=ThisDisplay.GetPointObjectFromName(St2)
        If Co = 1 Then 'Write to Excel
            ThisDisplay.g_Excel_Sheet.Range(St)=Point.Value
        Else 'Read From Excel
            Point.Value=ThisDisplay.g_Excel_Sheet.Range(St)
        End If
    Next X
End Sub
```

10. Return to the display, and create two radio buttons. Name one of them `Read from Excel` and the other `Write to Excel`. Configure both of them to run a VBA Script. Keep the execute.
11. Create a new macro for each of the two radio buttons. Call one macro `Rd` and call the other `Wr`. Both of these two macros will call the `Read_Write` procedure. The Code for each macro should look like this:

```
Sub Wr(o As GwxPick)
    Call ThisDisplay.Read_Write(1)
End Sub

Sub Rd(o As GwxPick)
    Call ThisDisplay.Read_Write(0)
End Sub
```

12. Go back to the display, and go into Runtime. Notice Excel starts up automatically with the value 0 in the A1 and A2 cells. If you click on the `Write to Excel` button and change the value of one of the process points in M-Graphics, you will notice the value in one of the cells in Excel changes. If you click on the `Read from Excel` button and change one of the two values in Excel, the process point will then update to match the value in Excel.

Troubleshooting

Table 12-5: Troubleshooting

Problem	Solution
An Error Parsing Data Source message appears after left or right mouse-click on a point data display which runs Visual Basic for Applications (VBA) in a standard point object or in a point dynamic not properly mapped during alias definition.	<p>The graphic contains a corrupted point object. Find this object and delete it. To delete the corrupted object from the display:</p> <ol style="list-style-type: none"> 1. Select Edit >Find and set the Type to Text Label. 2. For example, if <<N1.CSData.S>>.Present_Value is the corrupted information, in the Find What field, paste: <<N1.CSData.S>>.Present_Value. 3. Click Apply. 4. Select the one object that appears in the displayed tree diagram. 5. Click Show Selection. In the top left corner, the blue handles of the corrupted object appear. 6. Click Close. 7. Click Delete. 8. Click Save As to save the display under a different name. 9. Exit M-Graphics. 10. Restart M-Graphics. 11. Load the display in Runtime mode.
Multiple VBA Option Explicit statements appear in graphic files with VBA or duplicate VBA codes appear. The graphic shows a compile error in VBA when running. This occurs after templates have been edited and updated in the graphics.	<p>After editing or updating templates, delete any multiple Option Explicit statements using the VBA Editor. This problem is fixed at M-Web Release 2.0, M3 Workstation Release 2.0, or M5 Workstation Release 1.1 with M-Graphics Release 3.0. Earlier files with existing problems need to be edited to remove redundant code lines after upgrading to M-Graphics Release 3.0.</p> <p>Files produced at M-Graphics Release 3.0 do not have this problem.</p>

Chapter 13

M-Graphics ActiveX Control

Introduction

This chapter describes how to:

- insert M-Graphics ActiveX control
- configure M-Graphics ActiveX control

Key Concepts

M-Graphics ActiveX Control

M-Graphics ActiveX control (GWXview32.OCX) is capable of running M-Graphics displays. The advantage of ActiveX controls, like M-Graphics, is that they can be embedded into any control container (i.e., Visual Basic forms, Internet Explorer HyperText Markup Language (HTML) pages, M-Graphics displays).

M-Graphics ActiveX control is essentially a Runtime-only component (with some minimal configuration capability); the Runtime-only design allows the ActiveX to conserve memory. An M-Graphics ActiveX control executes displays created by gwx32.exe (M-Graphics).

Runtime Window Properties Mode

This feature of M-Graphics makes it possible to see what a display looks like in Runtime mode without actually entering Runtime. Pressing Ctrl + R toggles the display window between Configure and Runtime modes. When the action toolbars are hidden, the configuration functions can be found in the main menu or in the right-click pop-up menu.

Note: Pressing Ctrl + M toggles between Configuration and Runtime modes.

Procedure Overview

Table 13-1: M-Graphics ActiveX Control

To Do This	Follow These Steps:
Insert M-Graphics ActiveX Control	From the Edit menu, select Insert New Object. Select the item Johnson Controls M-Graphics Control. Click OK.
Configure M-Graphics ActiveX Control	Double-click the M-Graphics ActiveX control to display the control's Property page. Fill in the parameters. Click OK.

Detailed Procedures

Inserting M-Graphics ActiveX Control

Techniques for inserting an ActiveX control may vary slightly among different control containers; however, the basics are the same.

Note: If you are using the M-Graphics numeric keypad function to work with an M-Graphics ActiveX control, you must run the M-Graphics (gwx32.exe) as a standalone application on the M-Series Workstation. The numeric keypad function may not work within M-Graphics ActiveX controls.

To insert M-Graphics ActiveX control:

1. From the Edit menu, select Insert New Object. The Insert Object dialog box appears (Figure 13-1).

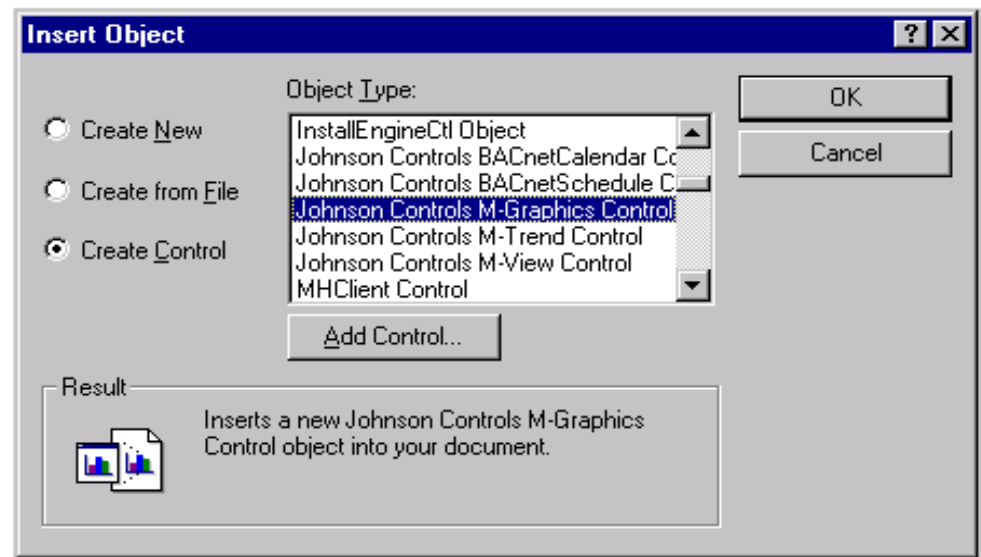


Figure 13-1: Insert Object Dialog Box

2. Select the item Johnson Controls M-Graphics Control.
3. Click OK. The M-Graphics ActiveX control appears in the center of the display.

Note: An M-Graphics ActiveX control may also be inserted by clicking on the icon in the ActiveX toolbar (Figure 13-2).



Figure 13-2: ActiveX Toolbar

Configuring M-Graphics ActiveX Control

To configure M-Graphics ActiveX control:

1. Double-click the M-Graphics ActiveX to display the Property page shown in Figure 13-3.
2. Fill in the parameters described in Table 13-2.
3. Click OK.

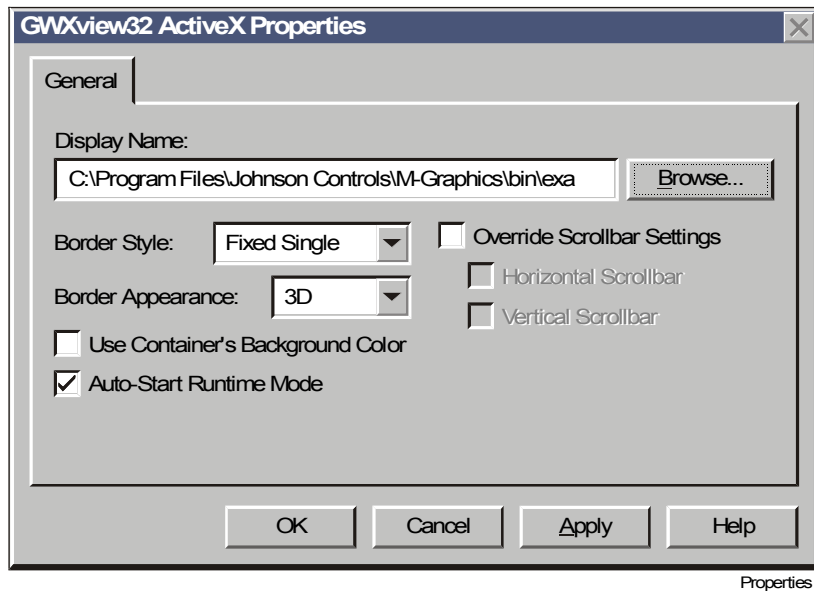


Figure 13-3: M-Graphics Property Page

Table 13-2: M-Graphics ActiveX Property Page

Parameter	Description
Display Name	Specifies the file name of the M-Graphics display to be loaded in this control. Use the Browse button to browse for display files.
Border Style	Specifies appearance of the window border: No Border or Fixed Single.
Border Appearance	Specifies appearance of border: Flat or 3-D.
Use Container's Background Color	Automatically sets the background color of the control to match the background color of the container in which this control is embedded. This property only works if the container supports the AmbientBackColor property. M-Graphics and Visual Basic forms both support this property.
Auto-Start Runtime Mode	Automatically enters Runtime mode when the control's container enters Runtime mode. This property only works if the container supports the AmbientUserMode property. M-Graphics and Visual Basic forms both support this property.
Override Scrollbar Settings	Indicates the control will override the scroll bar visibility settings of the display currently loaded in the control. When not checked, the control will use the scroll bar settings defined in the currently loaded display.
Horizontal Scrollbar	Determines if the horizontal scroll bar is visible. This property is ignored if Override Scrollbar Settings is not checked.
Vertical Scrollbar	Determines if the vertical scroll bar is visible. This property is ignored if Override Scrollbar Settings is not checked.

For more information about the properties of the M-Graphics ActiveX, see the *OLE Automation Reference* chapter in this manual specifically the section describing the GWXview32 object type.

Note: Convert is a standard function of ActiveX objects that allows the conversion of ActiveX objects to metafiles, bitmaps, icons, etc. However, this function has no purpose in M-Graphics ActiveX objects.

TWXView32 ActiveX Control

Introduction

The TWXView32 is an OPC Client application and provides realtime trending within the M-Graphics product family. This chapter describes how to:

- insert a TWXView32 control
- edit general parameters
- edit fonts parameters
- edit workspace parameters
- edit grids parameters
- edit ranges parameters
- edit details parameters
- edit time parameters
- edit rate parameters
- edit pens parameters
- edit settings parameters
- view about information
- edit pens in Runtime
- edit trend in Runtime
- edit period in Runtime
- toggle freeze/unfreeze in Runtime
- view statistics in Runtime
- edit the focus pen in Runtime
- move between samples in Runtime

Key Concepts

TWXView32 ActiveX Control

TWXView32 ActiveX can be used with third party OPC server applications to provide excellent trending features.

Some of TrendWorX32 Viewer features include:

- displaying realtime data
- managing a sophisticated multi-threaded data buffering to guarantee data integrity with support for customer configurable data collection rates
- independent data collection/screen refresh mechanism
- extensive Runtime configuration
- built-in buffering system
- support for custom imported data
- multilingual support
- interface to OPC realtime servers
- easy Visual Basic integration
- extensive realtime signal information (including signal quality)
- expression handling
- realtime statistical information with CSV file support
- user configured alarm levels for signals
- user defined global or relative signal ranges
- fixed location or scrolling grids
- embedded signal simulator
- multiple plotting type support: Time plot, XY-plot, logarithmic plot, strip chart, bar plot

TWXView32 ActiveX in M-Graphics

The TWXView32 ActiveX integrates easily within Visual C++ or Visual Basic container applications. Containers, such as M-Graphics provide toolbar support for inserting the TWXView32 control. Once inserted into a document or a form, the component can be resized and configured.

The TWXView32 ActiveX saves almost all of its configuration data within its own streamed file format with extensions *.v32. The user **must** give a valid file name before the data are serialized and saved. The TWXView32 uses a working directory defined in containers (such as M-Graphics). This represents the working directory where project files and configurations files are stored. TWXView32 utilizes this directory as a starting point for saving and loading its configuration files.

Graphics built with Iconics TWXView32 control running within a container, such as M3 Workstation or Internet Explorer, may cause frequent crashing while switching from Configure mode to Runtime mode. While in Runtime mode, double-clicking on the control and switching to Configure mode crashes M-Graphics.

As a workaround, we recommend while updating the graphics with TWXView32 control, run M-Graphics application standalone. The crash only occurs when M-Graphics is running within another container such as M3 Workstation or Internet Explorer.

Connections

TWXView32 can display data from realtime signals connected to OPC servers, calculated variables using expressions, simulated data, and custom data through OLE automation. Signal connections can be done in various ways:

- in Configuration mode through the pens property page
- in Runtime mode through the real pens button of the toolbar
- using OLE methods (such as AddRealTimePen) and VB Script/VBA
- using drag and drop (text format) in Runtime mode only

Once a signal is connected to a TWXView32 pen, the TWXView32 adds the pen to its buffering system manipulation and starts data collection and data display. Use the Tags button to connect pens to realtime OPC server signals.

TWXView32 in Configure and Runtime Modes

Within the M-Graphics container and in Visual Basic forms, the component is initially placed in Configure mode; the user can configure the TWXView32 ActiveX through the property pages. Once the application is committed to Runtime, the TWXView32 is activated, which enables its user interface (toolbar). To connect a realtime pen to a signal using drag and drop, the TWXView32 must be activated and in Runtime mode. The drag and drop object is of text type containing a single signal name.

Functions Not Supported in Runtime

The following functions are not supported in Runtime:

- replacing point names. (User uses delete/add, or within the VBA environment, uses the Form Load event for replacing point names.)
- changing of plot type of the trend
- modifying file name for saving module's data
- modifying the interface to the Timesettings object using OLE
- modifying X-axis of XY plots
- modifying file name for saving statistics
- modifying preference for global ranges
- modifying preference for shifting grids
- modifying preference for startup value
- modifying preference for restoring range values

Runtime TWXView32 Toolbar

The buttons toolbar is the Runtime user interface to the ActiveX. The toolbar offers the user access to various Runtime TWXView32 ActiveX functions and operations. To view the toolbar, double right-click on TWXView32 during Runtime.

Disable the toolbar in order to configure trend or to zoom.

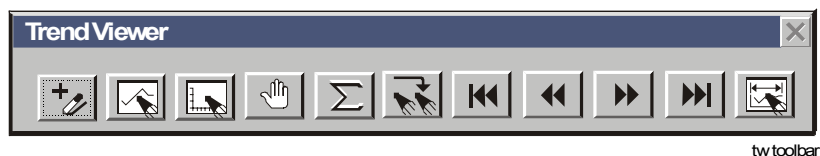


Figure 14-1: Runtime TWXView32 Toolbar

International Language Support

TWXView32 has all of its strings saved in the resource file, which can be modified to provide custom versions. In addition, it provides realtime support for loading a resource only DLL to support an international language. This DLL is compiled as a standard Win32 DLL, and is made up of all the resources and the ODL file of the TWXView32 module.

Asynchronous Downloading

The TWXView32 supports asynchronous downloading of configuration data when used inside ActiveX Document servers such as M-Graphics to produce documents.

To enable this feature on the General property pages, check the Use URL Path checkbox and supply a URL (Uniform Resource Locator) path. In case of viewing ActiveX documents within an Intranet setup, the user supplies a computer name and directory specifying the location of the TWXView32 configuration data (must be a shared directory) by using the convention:

`file:\\Computer Name\\Directory\\filename.V32`

Internal Architecture of ActiveX

TWXView32 allows the data to be collected at a different rate and graphically drawn at a multiple of the data collection rates in order to minimize overall system load. Currently minimal scan rate is set to be ten milliseconds, and it is required that all data collection rates set by the user be an integer multiple of ten milliseconds. In addition, the user configures the desired number of samples to view on the screen (up to a maximum of 3600) and the desired window time period. All objects exposed by the TWXView32 can be accessed directly through OLE automation, and their associated properties can be manipulated directly through C++ clients (which support OLE containment) or Visual Basic applications.

TWXView32 Object Capability

This section describes the objects exposed through OLE automation used in TWXView32. These are easily viewed in the Visual Basic Object Browser.

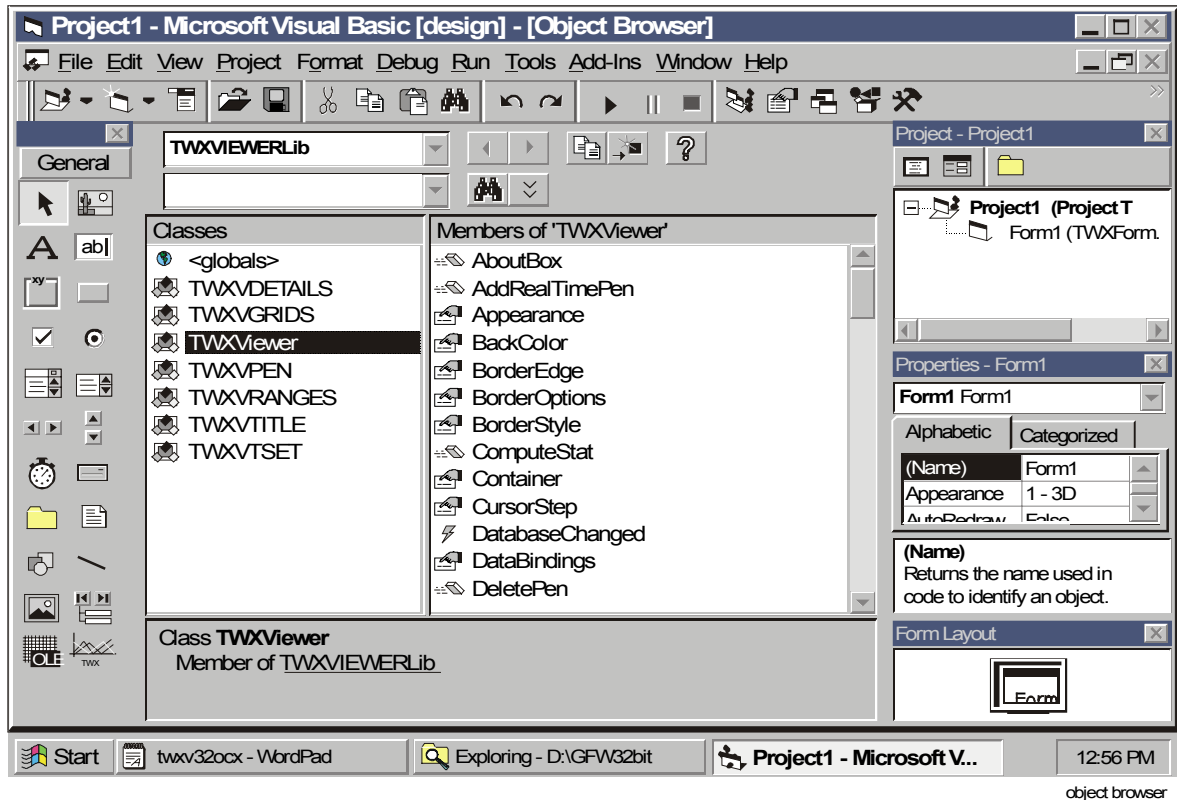


Figure 14-2: Object Browser

TWXView32 Properties

The TWXView32 ActiveX Properties dialog box is the configuration user interface to the ActiveX. Double-clicking anywhere in the ActiveX client during Configure mode opens the TWXView32 Properties dialog box (Figure 14-3). The dialog box fields are described in Table 14-1.

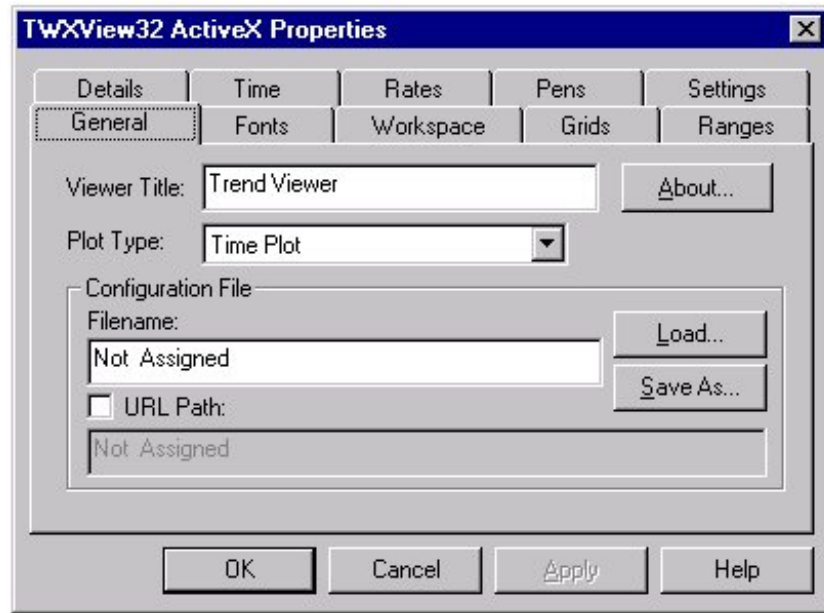


Figure 14-3: TWXView32 ActiveX Properties Dialog Box

Table 14-1: TWXView32 ActiveX Properties Dialog Box Tabs

Properties Page	Description
General	Default property page which handles: title, file selection, web configuration, plotting style.
Fonts	Determines font. The TWXView32 maintains four separate fonts: title, ranges, details, time font.
Workspace	Configures various aspects of the trend display such as showing range information, signal information, etc.
Grids	Manages the Grid Configuration for the trend display.
Ranges	Determines the ranges for the trend display. Depending on the selected plotting type, some options may or may not be enabled. Configure the number of visible pen ranges to be displayed.
Details	Manages the detail display information area and enabled choices closely match the selected plotting type. Configures the number of visible pen details to display.
Time	Configures the style of the time display inside the trend. Determines visible pen details.
Rates	Selects the desired data collection rate or number of samples.
Pens	Handles the signal connection, editing, and configuration for the data displayed in the trend.
Settings	Determines various settings for the control.
About Info	Lists copyright and version information.

OLE Automation

This section describes the OLE automation interface for TWXView32. Each ActiveX exposes a set of objects, which are used to configure and automate the control. Each object categorizes an area of functionality. Table 14-2 lists the objects exposed by the TWXView32 ActiveX.

Interfaces

- IOTitle
- IORanges
- IOTimeSettings
- IODetails
- IOPen
- IOGrids

The following properties and methods are categorized by the object in which they exist.

Methods/Properties for the TWXView32 ActiveX

Table 14-2: Properties for the TWXView32 ActiveX

Properties	Description
Boolean AutoStartRuntime	Autostarts the Runtime mode when the container application switches from Design to Active mode.
long ReadyState	Used for asynchronous download of configuration data.
LPDISPATCH Title	Dispatch interface to Title object
LPDISPATCH XGrids	Dispatch interface to X-Grids
LPDISPATCH YGrids	Dispatch interface to Y-Grids
LPDISPATCH Ranges	Dispatch interface to ranges
LPDISPATCH TimeSettings	Dispatch interface to time settings
LPDISPATCH Details	Dispatch interface to details
OLE_COLOR BackColor	Background color
OLE_COLOR DispColor	Trend Area color
OLE_COLOR ForeColor	Title color
Boolean ShowTitle	Determines if the main title is displayed. Default: True.
Boolean ShowRanges	Determines if the ranges field is displayed. Default: True.
Boolean ShowDetails	Determines if the details field is displayed. Default: True.
Continued on next page . . .	

Properties (Cont.)	Description
Boolean ShowTimeInfo	Determines if the time field is displayed. Default: True.
Boolean MainToolbar	Enables/disables button toolbar. Default: True.
BSTR PlotType	Determines PlotType.
BSTR BorderEdge	Determines trend area settings.
BSTR BorderOptions	Determines trend area settings.
LPFONTDISP TimeFont	Dispatch interface to the time font
LPFONTDISP DetailsFont	Dispatch interface to the detail font
LPFONTDISP MainFont	Dispatch interface to the main (title) font
LPFONTDISP RangesFont	Dispatch interface to the range font
short Appearance	Determines TWXView32 ActiveX appearance.
short BorderStyle	Determines TWXView32 ActiveX appearance.
Boolean FreezeMode	Enables/disables Freeze mode.
short NumberOfPens	Returns number of currently Configured pens.
BSTR FileName	Gets/sets file name for saving configuration data.
Boolean URLPathUsed	Enables/disables use of URL path for asynchronous downloads.
BSTR URLPathName	Gets/sets URL path name.
BSTR StatisticsFile	Gets/sets the CSV file used for statistical information storage.
Boolean StatFileAppend	Enables/disables Append mode for statistical file.
Boolean GlobalRanges	Enables/disables the use of global ranges. Default is True.
Boolean ShiftGrids	Enables/disables the shifting of Grids during Runtime. Default is True.
double StartUp	Gets/sets initial offset for plotting. Default is 0.
short CursorStep	Gets/sets the cursor step for stepping in Freeze mode.
Boolean RestoreRanges	Enables/disables restoring of ranges modified during Runtime.
long LanguageID	Gets/sets the language ID for loading a multilingual resource DLL.
Boolean FillBackColor	Enables/disables the color fill of trend display areas. This should be used prior and after printing (within the container) to conserve color ink on the printer.

Table 14-3: Methods for the TWXView32 ActiveX

Properties	Description
void Refresh()	Standard refresh method. Call after visual modifications.
Boolean AddRealTimePen (BSTR PenName)	Adds a realtime pen.
LPDISPATCHGetPen (BSTR PenName, BSTR PenType)	Returns Dispatch interface of pen. User must release interface when done. Valid types are Real Time, History.
Boolean SetXAxis (BSTR PenName)	Used in XY-Plot to select a pen for X-axis.
VARIANT GetPenNames()	Returns a VARIANT String array of all pen names.
Boolean DeletePen (BSTRPenName, BSTR PenType)	Deletes a pen. Valid types are Real Time, History.
Boolean SetDetailsFromPen (BSTR PenName, BSTR PenType)	Populates details info from the pen. Valid types are Real Time, History.
Boolean EnterAnimate()	Enters Runtime (Animate) mode.
Boolean ExitAnimate()	Exits Runtime (Animate) mode.
Boolean GetAnimateMode()	Returns True if Viewer is currently in Animate mode. Otherwise False is returned.
void AboutBox()	Displays the viewer's About box.
Boolean ComputeStat()	Computes and stores statistical information.
void SetHiRangeString (BSTR RangeString)	Sets point name for high range of Range Object.
Boolean ReplacePoint (BSTR PointName, BSTROldString, BSTRNewString, Boolean DoPenText)	Use to replace point names and pen descriptions if desired.
void SetPointValue (BSTR PointName, double PointValue, DATE PointTDate)	Use to feed custom data for an assigned pen.
void SetPenHiRangeString (BSTR PenName, BSTR RangeString, BSTR PenType)	Sets point name for high range of a pen.
void SetPenLoRangeString (BSTR PenName, BSTR RangeString, BSTR PenType)	Sets point name for low range of a pen.
void ChangeWinPeriod (long NewPeriod)	Changes trend period.
Boolean SetTrendRightTDate (DATE RightTDate0)	In Freeze mode, sets the trend's right time and date.

Events Fired

- ReadyStateChange
- InAnimateMode
- UnableToCreateThread
- OutOfAnimateMode
- DatabaseChanged
- ErrorInLoadingData

Not all methods/properties are supported during Runtime or Animation mode.

Methods/Properties for Supporting Objects**Table 14-4: Title Object Properties**

Properties	Description
BSTR TitleText	The main title of the viewer ActiveX
Boolean ShowTitle	Determines whether or not the main title is displayed. Default: True.

Table 14-5: TimeSettings Object Properties

Properties	Description
BSTR BorderEdge	Visual appearance of time display
BSTR BorderOptions	Visual appearance of time display
BSTR DateFormat	American/European
OLE_COLOR TextColor	Color for text displayed
long DataUpdatePeriod	Data collection rate
long DataTotalPeriod	Total time period of data
long Samples	Number of samples to display
OLE_COLOR FillColor	Color for filling time area
long DataDisplayMult	Gets/sets drawing intervals relative to data collection.
Boolean ShowDate	Determines if date information is displayed in the Viewer's time field. Default: True.
BSTR Size	Size for time display. It can be large, normal, small.
Boolean ShowTimeInfo	Determines if the time field is displayed. Default: True.
Boolean ShowMilliSecs	Shows millisecond divisions on time axis. Default: False.
DataUpdatePeriod, DataTotalPeriod, Samples	Set operations goes into effect the next time the TWXView32 has entered Runtime.

Table 14-6: Ranges Object Properties

Properties	Description
double High	Gets/sets high range value.
double Low	Gets/sets low range value.
BSTR BorderEdge	Visual appearance of range display
BSTR BorderOptions	Visual appearance of range display
BSTR Position	Gets/sets position of range display. It can be Right, Left.
BSTR NumberFormat	Gets/sets number format for range values.
Boolean ShowRanges	Determines if the range field is displayed. Default: True.
OLE_COLOR FillColor	Gets/sets fill color of range display.
OLE_COLOR TextColor BSTR Size	Size for Time display. It can be large, normal, small.
double XHigh	Gets/sets high range value for X-axis in XY-plots.
double XLow	Gets/sets low range value for X-axis in XY-plots.
BSTR XNumberFormat	Gets/sets number format for X-axis range values in XY-plots.
Boolean OverHighRange	Get/set override for high range. If set True, high range value will be computed from a Tag or Expression.
Boolean OverLowRange	Get/set override for low range. If set True, low range value will be computed from a Tag or Expression.
Short Visible Pens	Gets/sets the number of visible pen ranges on the ranges display area (if no Global ranges option is selected).

Table 14-7: Pen Object Properties

Properties	Description
OLE_COLOR PenColor	Gets/sets the color associated with the pen.
BSTR PenStyle	Gets/sets pen style such as Solid, Dotted, Dashed, etc.
short PenWidth	The width of the pen in logical units. Default: 1.
BSTR PenText	Pen Description
BSTR PenUnits	Pen Engineering Units
Boolean PenVisible	Determines if a pen is to be drawn.
BSTR PenPointName	Pen signal point name
double HighLimit	High limit value
double LowLimit	Low limit value
Boolean PenXAxis	Gets/sets a pen an X-axis in XY-plots.
double AlarmHigh	High alarm value for pen
double AlarmLow	Low alarm value for pen
double AlarmHiHi	High-high alarm value for pen
double AlarmLoLo	Low-low alarm value for pen
Boolean EnableAlarmLines	Enables drawing of alarm lines
long NumberOfSamples	Gets number of samples for the pen.
Boolean OverHighRange	Get/set override for high range. If set True, high range value will be computed from a Tag or Expression.
Boolean OverLowRange	Get/set override for low range. If set True, low range value will be computed from a Tag or Expression.
Boolean Stairplot	Enables stepwise plotting mode.

Table 14-8: Details Object Properties

Properties	Description
BSTR BorderEdge	Visual appearance of details
BSTR BorderOptions	Visual appearance of details
OLE_COLOR FillColor	The background color of the details field.
Boolean ShowPenPoint	Determines if a pens information is displayed.
Boolean ShowPenText	Determines if a pens text is displayed.
Boolean ShowPenUnits	Determines if a pens units is displayed.
Boolean ShowPenValue	Determines if a pens value is displayed.
Boolean ShowPenHiLo	Determines if a pens Hi/Lo information is displayed.
Boolean ShowPenHiLoAlarm	Determines if a pens Hi/Lo Alarm information is displayed.
Boolean ShowPenHiHiLoLoAlarm	Determines if a pens HiHi/LoLo Alarm information is displayed.
Boolean ShowPenValidity	Determines if signal quality will be displayed.
Boolean ShowXAxis	Determines if X-axis values are shown for XY-plots.
Boolean ShowDetails	Determines if the details field is displayed. Default: True.
OLE_COLOR TextColor	Text color of all text in the details field.
BSTR Size	Gets/sets the size of the details area.
BSTR NumberFormat	Gets/sets the number format of the pen.
Boolean ShowPenTime	Determines if a pens time information is displayed.
Boolean ShowPenDate	Determines if a pens date information is displayed.
BSTR DateFormat	Gets/sets American or European date format.
Short Visible Pens	Gets/sets the number of visible pen details to be drawn in the details display area.

Table 14-9: Grids Object Properties

Properties	Description
Boolean ShowGrids	Determines if grids are displayed in the Viewer's main display area. Default: True.
OLE_COLOR GridColor	The color of the grid
short GridWidth	The grid width in logical units
BSTR GridStyle	Gets/sets grid style (such as Solid, Dotted, Dashed, etc.).
short GridNumber	Gets/sets grid number.

Procedure Overview

Table 14-10: TWXView32 ActiveX Control

To Do This	Follow These Steps:
Insert a TWXView32 Control	In M-Graphics on the Edit menu, select Insert New Object. Select Create Control and ICONICS TWXView32 for the object type. Click OK.
Edit General Parameters	Double-click on the TWXView32 control. Select the General tab. Modify the parameters.
Edit Fonts Parameters	Double-click on the TWXView32 control. Select the Fonts tab. Modify the parameters.
Edit Workspace Parameters	Double-click on the TWXView32 control. Select the Workspace tab. Modify the parameters.
Edit Grids Parameters	Double-click on the TWXView32 control. Select the Grid tab. Modify the parameters.
Edit Ranges Parameters	Double-click on the TWXView32 control. Select the Ranges tab. Modify the parameters.
Edit Details Parameters	Double-click on the TWXView32 control. Select the Details tab. Modify the parameters.
Edit Time Parameters	Double-click on the TWXView32 control. Select the Time tab. Modify the parameters.
Edit Rate Parameters	Double-click on the TWXView32 control. Select the Rate tab. Modify the parameters.
Edit Pens Parameters	Double-click on the TWXView32 control. Select the Pens tab. Modify the parameters.
Edit Settings Parameters	Double-click on the TWXView32 control. Select the Settings tab. Modify the parameters.
View About Information	Double-click on the TWXView32 control. Select the About tab.
Edit Pens in Runtime	In Runtime, double-click in the display to open the TWXView32 Runtime toolbar. Click on the Edit Pen button. Edit the parameters.
Edit Trend in Runtime	In Runtime, double-click in the display to open the TWXView32 Runtime toolbar. Click on the Edit Trend button. Edit the parameters.
Edit Period in Runtime	In Runtime, double-click in the display to open the TWXView32 Runtime toolbar. Click on the Edit Period button. Edit the parameters.
Toggle Freeze/Unfreeze in Runtime	In Runtime, double-click in the display to open the TWXView32 Runtime toolbar. Click on the Freeze button to toggle between freeze and unfreeze.
View Statistics in Runtime	In Runtime, double-click in the display to open the TWXView32 Runtime toolbar. Click on the Statistics button.
Edit the Focus Pen in Runtime	In Runtime, double-click in the display to open the TWXView32 Runtime toolbar. Click on the Focus Pen button. Edit the options.
Move Between Samples in Runtime	In Runtime, double-click in the display to open the TWXView32 Runtime toolbar. Click on one of the following buttons: Page Left, Cursor Left, Cursor Right, or Page Right.

Detailed Procedures

Inserting a TWXView32 Control

To insert a TWXView32 control:

1. In M-Graphics on the Edit menu, select Insert New Object. The Insert Object dialog box appears (Figure 14-4).

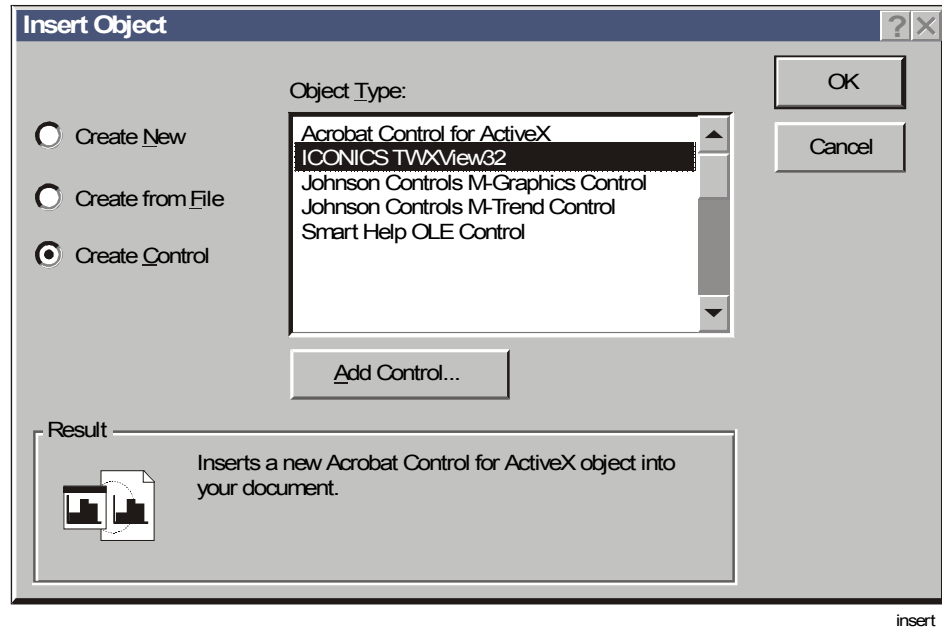



Figure 14-4: Insert Object Dialog Box

2. Select Create Control and ICONICS TWXView32 for the object type.
3. Click OK. A TWXView32 control is added to the display (Figure 14-5).

Note: As a shortcut, insert a TWXView32 ActiveX control by clicking on the  icon in the ActiveX toolbar.

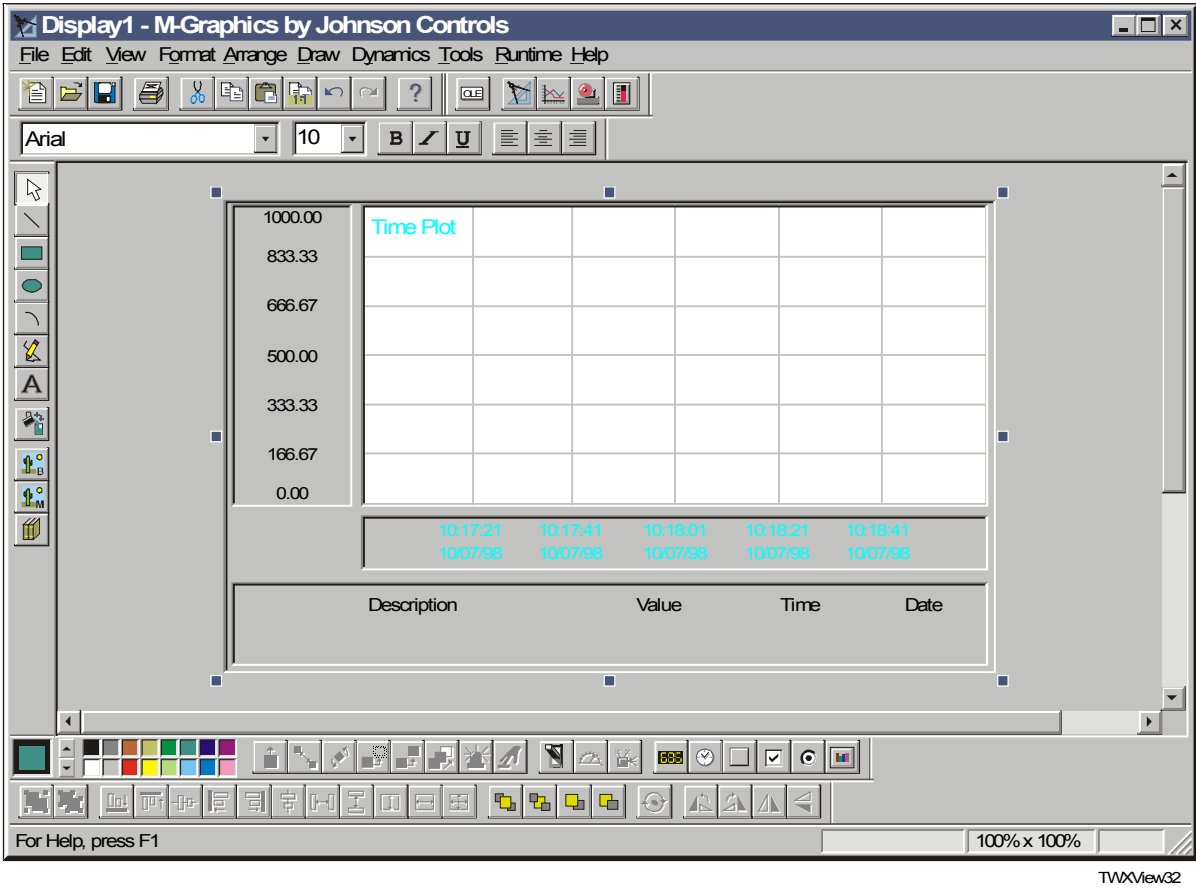


Figure 14-5: TWXView32 Control in M-Graphics Display

Editing General Parameters

To edit general parameters:

1. Double-click on the TWXView32 control. The TWXView32 ActiveX Properties dialog box appears (Figure 14-6).



Figure 14-6: ActiveX Properties Dialog Box: General Tab

2. Select the General tab.
3. Modify the parameters using Table 14-11.

Note: Choose the plotting style first, because this choice determines which options are available.

Table 14-11: General Tab Parameters

Parameter	Description
Viewer Title	Displays title of the display.
Plot Type	Determines available plot types: <ul style="list-style-type: none"> • Time Plot - plots variables vs. time • XY Plot - plots variable vs. variable • Bar Plot - plots a variable plot bar form • Logarithmic Plot - plots variables relative to the value of another variable logarithmically • Strip Chart - plots variables vs. time; however, this window plots data vertically. With strip charts, you can only display ranges for two pens in the Range window.
Filename	Displays name of current file with .v32 extension.
Load	Loads a new configuration file over an existing one.
Save As	Saves the configuration data in a file with a .v32.
URL Path	If checked, saves the trend window to a URL. In addition, the user can load a new configuration file over an existing file by using the Load button.

Editing Fonts Parameters

To edit fonts parameters:

1. Double-click on the TWXView32 control. The TWXView32 ActiveX Properties dialog box appears.
2. Select the Fonts tab (Figure 14-7).

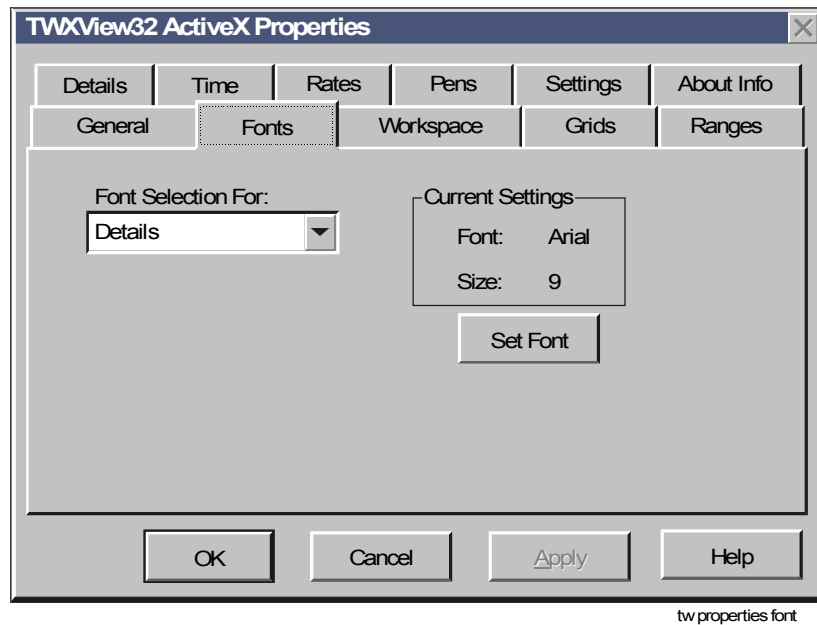


Figure 14-7: TWXView32 ActiveX Properties Dialog Box: Fonts Tab

3. Modify the parameters using Table 14-12.

Table 14-12: Fonts Tab Parameters

Parameter	Description
Font Selection For	Determines for which section the font is set: details, ranges, time settings, or title.
Set Font	Opens Font dialog box to specify font, style, point size, effects, and script.

Editing Workspace Parameters

To edit workspace parameters:

1. Double-click on the TWXView32 control. The TWXView32 ActiveX Properties dialog box appears.
2. Select the Workspace tab (Figure 14-8).

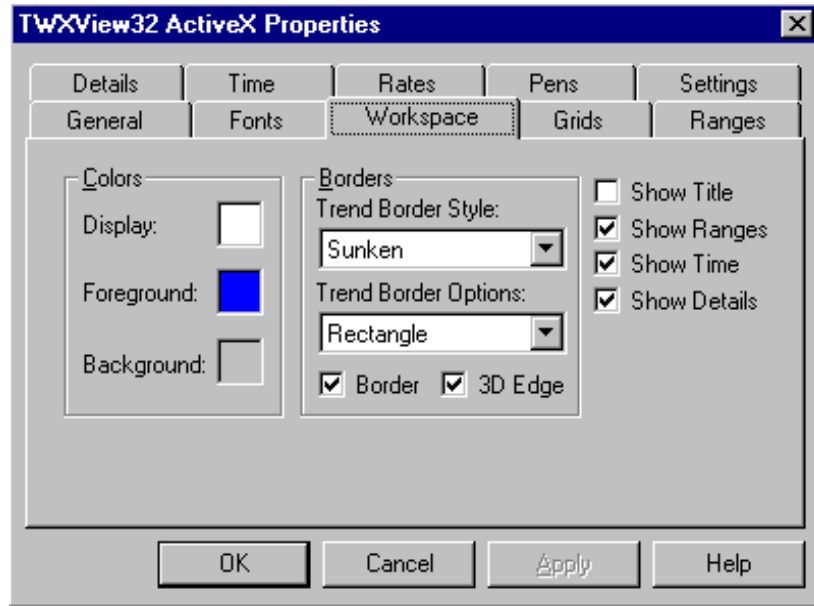


Figure 14-8: TWXView32 ActiveX Properties Dialog Box: Workspace Tab

3. Modify the parameters using Table 14-13.

Table 14-13: Workspace Tab Parameters

Parameter	Description
Colors	Determines background, foreground, and display color.
Trend Border Style	Determines the type of border: bumped, etched, raised, or sunken.
Trend Border Options	Determines border placement: bottom, bottom left, bottom right, left, rectangle, right, top, top left, or top right.
Border	Toggles between invisible and visible border.
3D Edge	Toggles between invisible and visible 3D edge.
Show Title	Show/hide title of graphic.
Show Ranges	Show/hide ranges of graphic.
Show Time	Show/hide time.
Show Details	Show/hide details of graphic.

Editing Grids Parameters

The Grids tab allows the user to configure the type, style, size, and color of the display grid. The Show grid box makes the grid visible, and the Snap To Grid button makes all borders of the display objects align with the grid lines.

To edit grids parameters:

1. Double-click on the TWXView32 control. The TWXView32 ActiveX Properties dialog box appears.
2. Select the Grids tab (Figure 14-9).

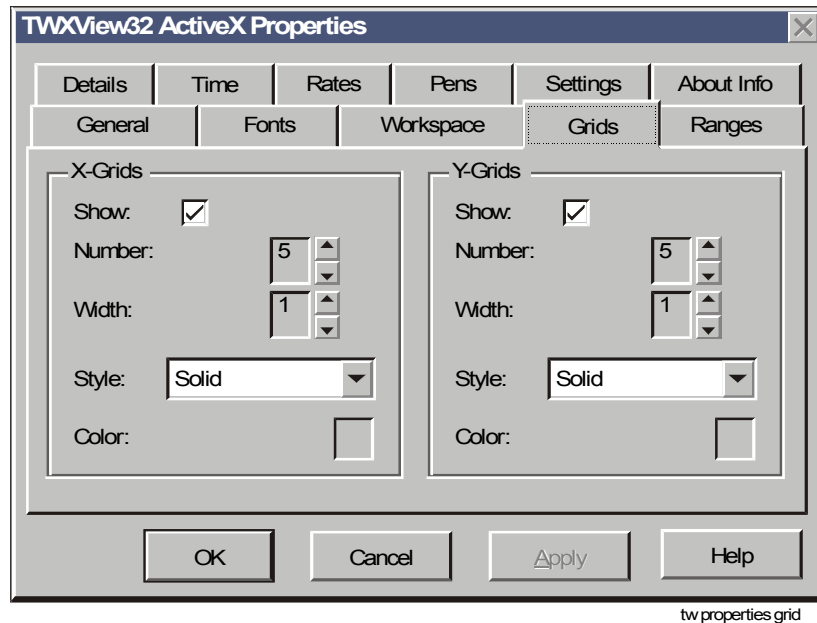


Figure 14-9: TWXView32 ActiveX Properties Dialog Box: Grids Tab

3. Modify the parameters using Table 14-14.

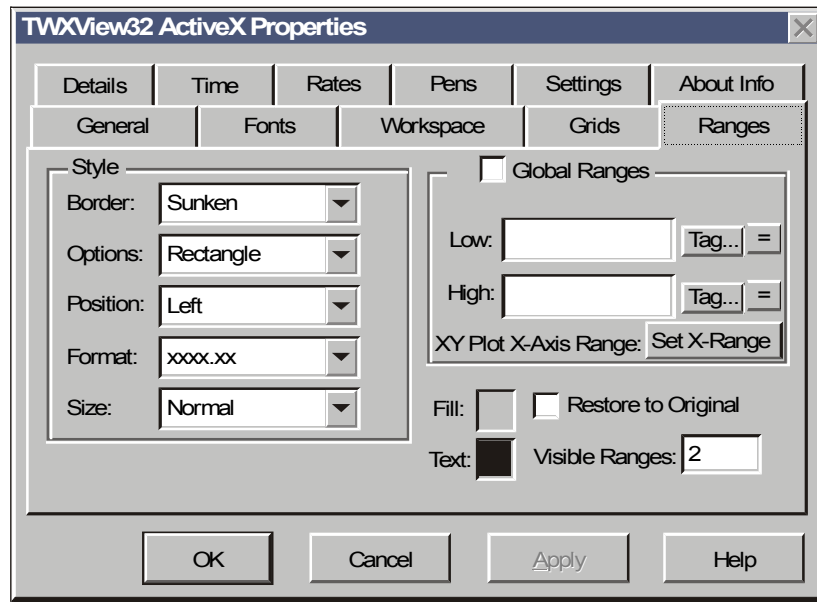
Table 14-14: Grids Tab Parameters

Parameter	Description
Show	Shows/hides grid.
Number	Specifies number of lines for each axis.
Width	Determines line width.
Style	Determines line style: dashed, dotted, dotted dashed, or solid.
Color	Specifies grid color.

Editing Ranges Parameters

To edit ranges parameters:

1. Double-click on the TWXView32 control. The TWXView32 ActiveX Properties dialog box appears.
2. Select the Ranges tab (Figure 14-10).



tw properties ranges

Figure 14-10: TWXView32 ActiveX Properties Dialog Box: Ranges Tab

3. Modify the parameters using Table 14-15.

Table 14-15: Ranges Tab Parameters

Parameter	Description
Border	Determines the type of border: bumped, etched, raised or sunken.
Options	Determines range placement: bottom, bottom left, bottom right, left, rectangle, right, top, top left or top right.
Position	Determines ranges field placement in the Trend window: right or left.
Format	Determines how the numbers representing the ranges appear: xxx.xx, xxxx, xxxx.xx.
Size	Describes the size of the entire ranges section: large, normal, or small.
Global Ranges	Enable/disable global ranges and change the number of visible pens.
Low	Specifies low range.
High	Specifies high range.
Set X-Range	If you are using an XY plot, X-Ranges allows the user to set the value and format of the low and high X-Ranges.
Fill	Determines fill color.
Text	Determines text color.
Restore to Original	Maintains the ranges even if the user changes something in Runtime.
Visible Ranges	Determines number of visible ranges.

Editing Details Parameters

To edit details parameters:

1. Double-click on the TWXView32 control. The TWXView32 ActiveX Properties dialog box appears.
2. Select the Details tab (Figure 14-11).

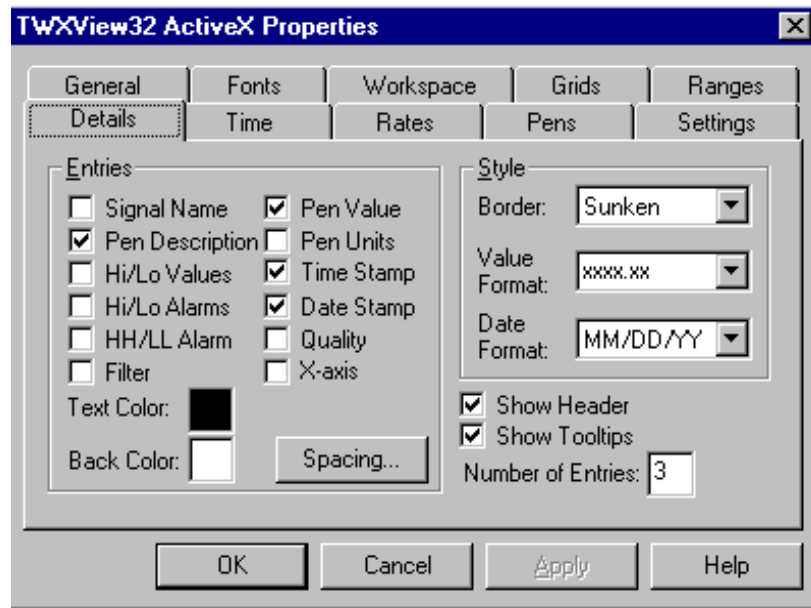


Figure 14-11: TWXView32 ActiveX Properties Dialog Box: Details Tab

3. Modify the parameters using Table 14-16.

Table 14-16: Details Tab Parameters

Parameter	Description
Entries	Determines what entries appear in Runtime mode.
Text Color	Determines the color of the text.
Back Color	Determines the color the background appears.
Style	Determines the style of object.
Border	Determines the type of border: bumped, etched, raised, or sunken.
Value Format	Determines details placement: bottom, bottom left, bottom right, left, rectangle, right, top, top left, or top right.
Date Format	Determines the appearance of the month, date, and year.
Show Header	Shows/hides the header.
Show ToolTips	Shows/hides the ToolTips.
Number of Entries	Determines how many entries appear on graphic.

Editing Time Parameters

To edit time parameters:

1. Double-click on the TWXView32 control. The TWXView32 ActiveX Properties dialog box appears.
2. Select the Time tab (Figure 14-12).

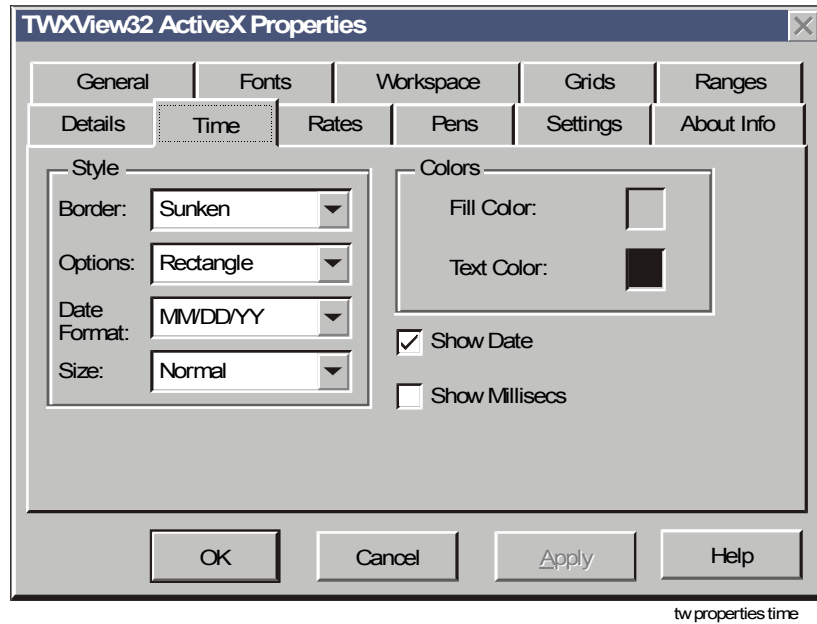


Figure 14-12: TWXView32 ActiveX Properties Dialog Box: Time Tab

3. Modify the parameters using Table 14-17.

Table 14-17: Time Tab Parameters

Parameter	Description
Border	Determines the type of border: bumped, etched, raised, or sunken.
Options	Determines time placement: bottom, bottom left, bottom right, left, rectangle, right, top, top left, or top right.
Date Format	Determines appearance of dates: MM/DD/YY or DD/MM/YY.
Size	Describes the size of the entire size section: large, normal, or small.
Fill Color	Determines fill color.
Text Color	Determines text color.
Show Date	Hides/shows date.
Show Millisecs	Hides/shows milliseconds.

Editing Rate Parameters

To edit rate parameters:

1. Double-click on the TWXView32 control. The TWXView32 ActiveX Properties dialog box appears.
2. Select the Rate tab (Figure 14-13).

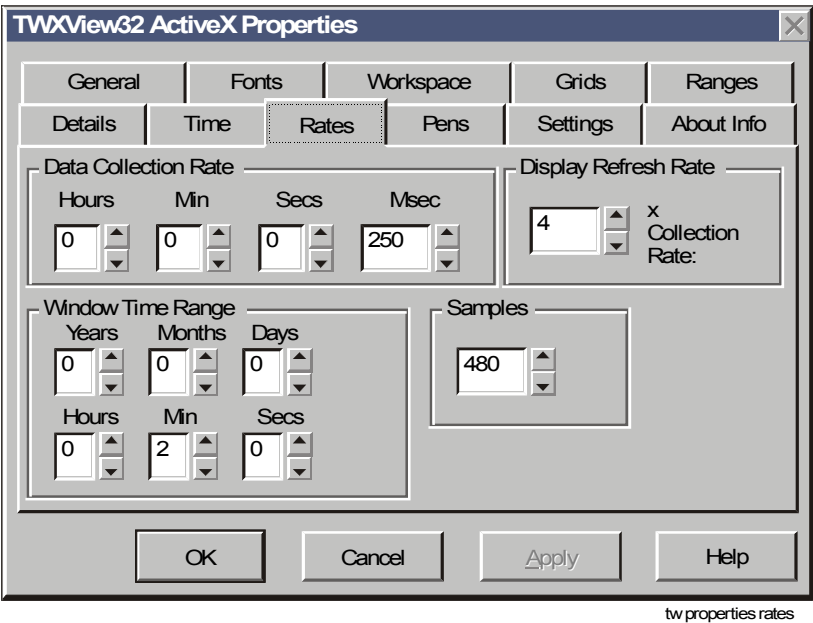


Figure 14-13: TWXView32 ActiveX Properties Dialog Box: Rate Tab

3. Modify the parameters using Table 14-18.

Table 14-18: Rate Tab Parameters

Parameter	Description
Data Collection Rate	Determines how often data is collected per day.
Display Refresh Rate	Determines how often the display refreshes.
Window Time Range	Determines number of days data is collected.
Samples	Determines number of samples collected.

Editing Pens Parameters

Editing pens is available in Configure and Runtime modes.

1. Double-click on the TWXView32 control. The TWXView32 ActiveX Properties dialog box appears.
2. Select the Pens tab (Figure 14-14).

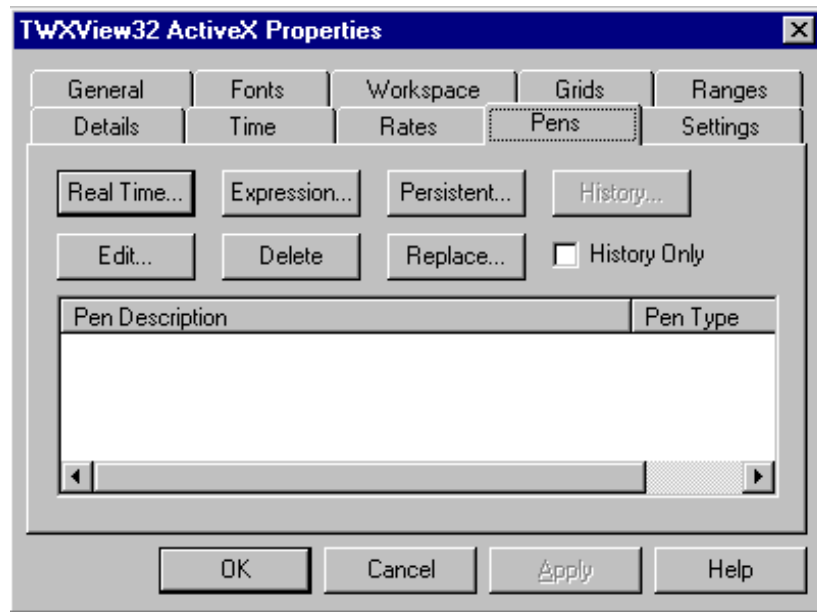


Figure 14-14: TWXView32 ActiveX Properties Dialog Box: Pens Tab

3. Modify the parameters using Table 14-19.

Note: The maximum number of user-configurable pens is 32.

Table 14-19: Pen Tab Parameters

Parameter	Description
Real Time	Opens the Runtime mode.
Expression	Opens the Edit Expressions dialog box. Refer to the <i>Adding Dynamics</i> chapter for details on the expressions.
Persistent	Allows a trend to be maintained when the graphic containing the trend is not active.
History	Opens the Persistent Trend dialog box listing previously used tags.
Edit	Refer to the <i>Editing Pens in Runtime</i> section in this chapter for a description of the options.
Delete	Deletes current Tag.
Replace	Opens the Search and Replace dialog box. Refer to the <i>Editing Objects</i> chapter for details on the searching and replacing.

Editing Settings Parameters

To edit settings parameters:

1. Double-click on the TWXView32 control. The TWXView32 ActiveX Properties dialog box appears.
2. Select the Settings tab (Figure 14-15).

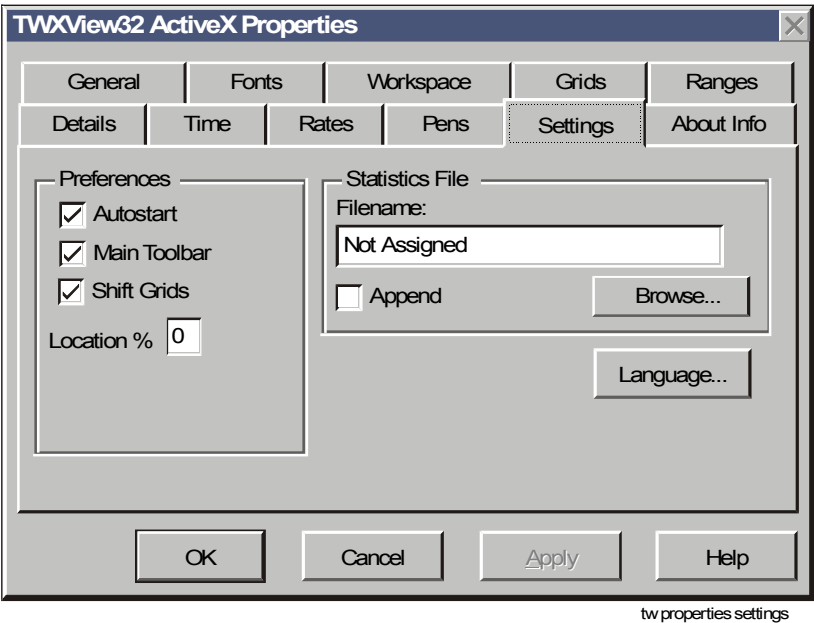


Figure 14-15: TWXView32 ActiveX Properties Dialog Box: Settings Tab

3. Modify the parameters using Table 14-20.

Table 14-20: Settings Tab Parameters

Parameter	Description
Preferences	Preferences include: <ul style="list-style-type: none"> • Autostart: Automatically places TWXView32 in Runtime • Main Toolbar: Show/hide main toolbar • Shift Grids: Enable Disable shifting of Grids
Filename	Specifies CSV file for saving statistical information during Runtime.
Append	Determines if you add information to current file or overwrite current file.
Browse	Opens a dialog box to search for specific files.
Language	Specifies language.

Viewing About Information

To view about information:

1. Double-click on the TWXView32 control. The Iconics About Box dialog box appears (Figure 14-16).

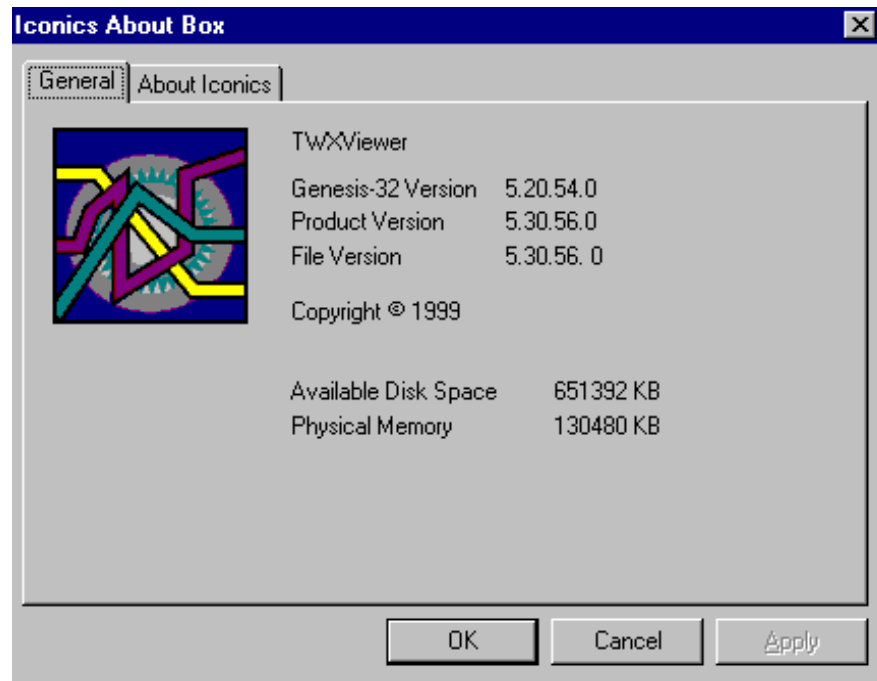



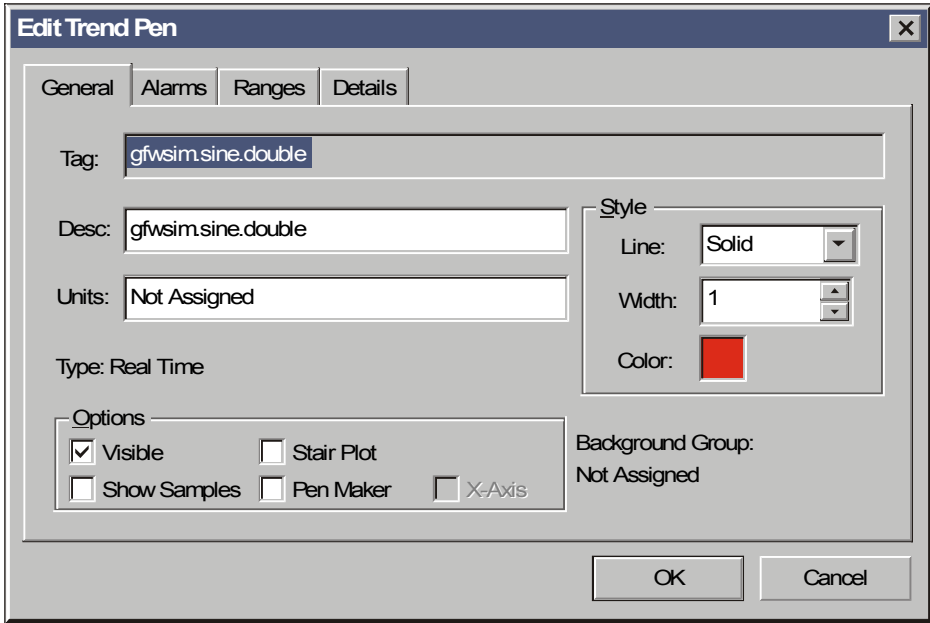
Figure 14-16: Iconics About Box Dialog Box

2. Select the General tab.

Editing Pens in Runtime

To edit pens parameters in Runtime:

1. In Runtime, double-click in the display to open the TWXView32 Runtime toolbar.
2. Click on the Edit Pen button .
3. Select a pen and click Edit. The Edit Trend Pen dialog box appears (Figure 14-17).
4. Edit the parameters. Refer to Table 14-21 for details.



Edit Pen General


Figure 14-17: Edit Trend Pen in Runtime: General Tab

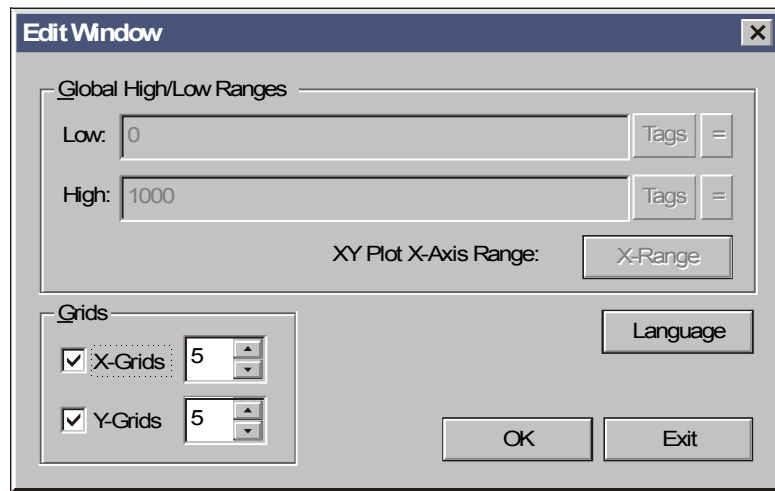
Table 14-21: Edit Trend Pen Parameters

Tab	Parameter	Description
General	Tag	Displays current Tag.
	Description	Describes current Tag.
	Units	Displays current units.
	Options	Enables/disables following options: showing the pen, using a stair plot, showing samples, using pen marker.
	Style	Determines line style, line width and color.
Alarms	Low	Lists alarm low limit.
	High	Lists alarm high limit.
	LowLow	Lists another alarm low limit.
	HighHigh	Lists another alarm high limit.
	Alarm Lines	Shows/hides alarm lines.
Ranges	Low	Displays low range.
	High	Displays high range.
	Color	Determines range fill and text color.
	Format	Determines range format.
	Autoscale	Enables/disables range autoscale.
Details	Fill Color	Determines fill color.
	Text Color	Determines text color.
	Format	Determines format.

Editing Trend in Runtime

To edit trend in Runtime:

1. In Runtime, double-click in the display to open the TWXView32 Runtime toolbar.
2. Click on the Edit Trend button . The Edit Window dialog box appears (Figure 14-18).
3. Edit the parameters. Refer to Table 14-22 for details.



Edit Window


Figure 14-18: Edit Window Dialog Box

Table 14-22: Edit Window Parameters

Parameter	Description
Global High/Low Ranges	Display ranges assigned in the Ranges tab. If global ranges was unchecked, this option is unavailable.
Grids	Determines number of grids displayed.
Language	Determines language.

Editing Period in Runtime

To edit period in Runtime:

1. In Runtime, double-click in the display to open the TWXView32 Runtime toolbar.
2. Click on the Edit Period button . The Trend Edit Period dialog box appears (Figure 14-19).
3. Edit the parameters. Refer to Table 14-23 for details.

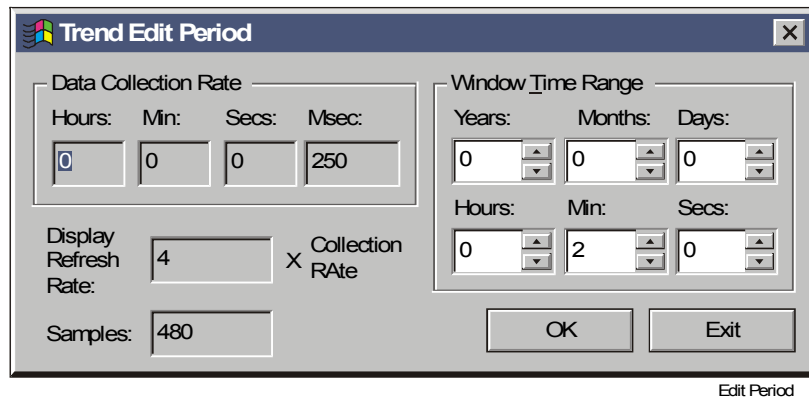



Figure 14-19: Trend Edit Period Dialog Box

Table 14-23: Edit Trend Period Parameters

Parameter	Description
Data Collection Rate	Determines how often data is collected per day.
Display Refresh Rate	Determines how often the display refreshes.
Window Time Range	Determines number of days data is collected.
Samples	Determines number of samples collected.


Toggling Freeze/Unfreeze in Runtime

To toggle freeze/unfreeze in Runtime:

1. In Runtime, double-click in the display to open the TWXView32 Runtime toolbar.
2. Click on the Freeze button  to toggle between freeze and unfreeze.

Viewing Statistics in Runtime

To view statistics in Runtime:

1. In Runtime, double-click in the display to open the TWXView32 Runtime toolbar.
2. Click on the Statistics button . The Trend Statistics dialog box appears (Figure 14-20) and lists current trend statistics.

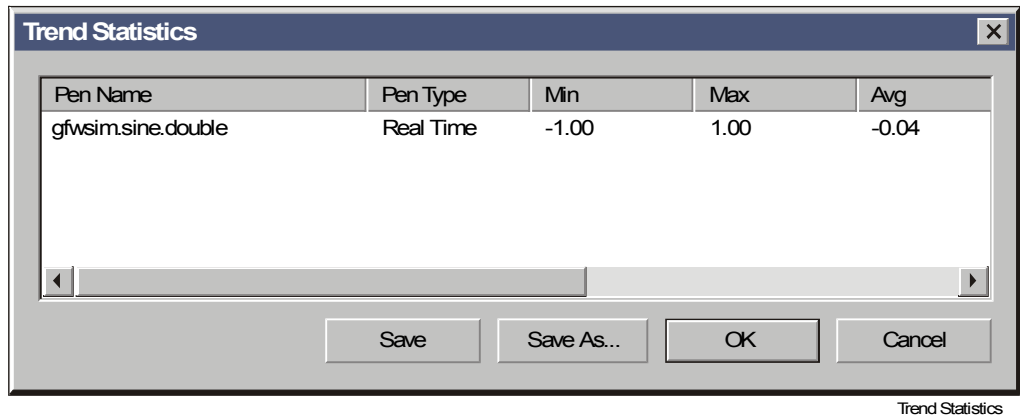



Figure 14-20: Trend Statistics Dialog Box

Editing the Focus Pen in Runtime

To edit the focus pen in Runtime:

1. In Runtime, double-click in the display to open the TWXView32 Runtime toolbar.
2. Click on the Focus Pen button . The Focus Pen(s) Selection dialog box appears (Figure 14-21).
3. Edit the options.

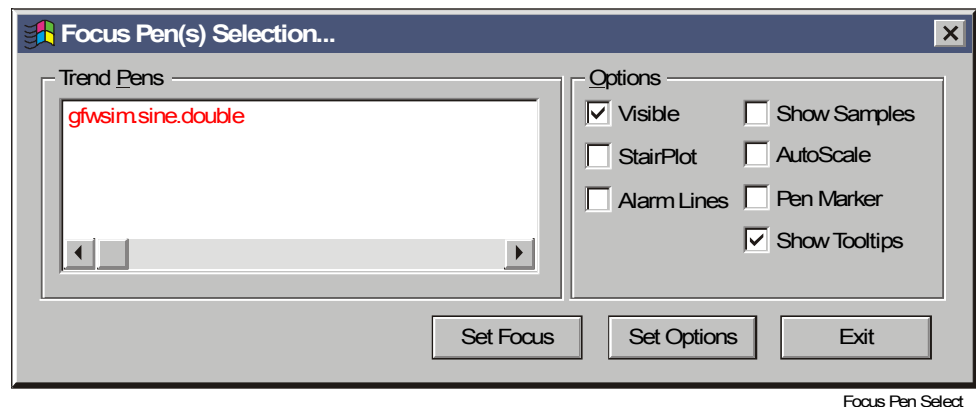






Figure 14-21: Focus Pen Selection Dialog Box

Moving Between Samples in Runtime

To move between samples in Runtime:

1. In Runtime, double-click in the display to open the TWXView32 Runtime toolbar (only available in Freeze mode).
2. Click on one of the following buttons:  (page left),  (cursor left),  (cursor right), or  (page right).
3. Hold the Shift key while pressing the cursor right/left buttons moves the cursor left or right ten samples default.

GWXGauge ActiveX Control

Introduction

The GWXGauge is an ActiveX control that can be inserted into the M-Graphics application to display analog data. This chapter describes how to:

- insert a GWXGauge ActiveX control
- edit general parameters
- edit background parameters
- edit caption parameters
- edit bar parameters
- edit needle parameters
- edit value parameters
- edit scale1 parameters
- edit scale2 parameters
- edit fonts parameters
- edit slider/dial parameters
- edit warning zones parameters
- view about information

Key Concepts

GWXGauge ActiveX Control

The GWXGauge ActiveX is an ActiveX control that can be inserted in the M-Graphics application. The GWXGauge ActiveX displays and modifies data in a virtual instrument panel format on graphics. Use the GWXGauge ActiveX to display analog data with color and alarm settings. The GWXGauge ActiveX supports connection to OPC compliant servers and provides excellent capabilities for managing data.

GWXGauge ActiveX Properties

The GWXGauge ActiveX Properties dialog box is the configuration interface to the GWXGauge ActiveX. Double-clicking anywhere in the GWXGauge ActiveX client during Configure mode opens the GWXGauge ActiveX Properties dialog box (Figure 15-1). The dialog box fields are described in Table 15-1.

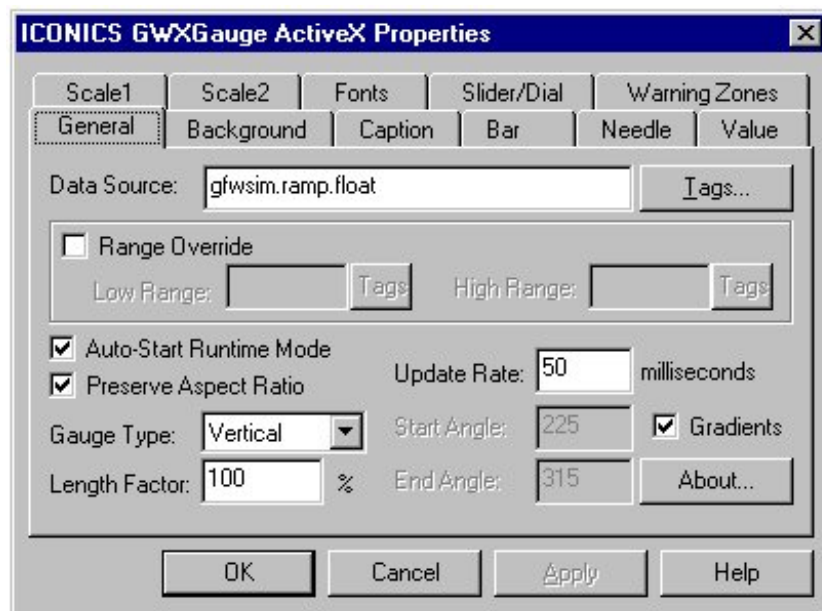


Figure 15-1: GWXGauge ActiveX Properties Dialog Box

Table 15-1: GWXGauge ActiveX Properties Dialog Box

Properties Page	Description
General	Allows user to select the tag used as the data source for the gauge.
Background	Configures gauge appearance within the container.
Caption	Configures the caption for the gauge and the text for the ToolTip.
Bar	Configures the color and style of the bar for a horizontal or vertical gauge.
Needle	Configures the color and style of the needle for a circular gauge.
Value	Configures the format and style of the text box at the bottom of the gauge.
Scale1	Configures the appearance of the interior, left, or bottom scale on the gauge.
Scale2	Configures the appearance of a second interior, left, or bottom scale on the gauge.
Fonts	Determines the font, font style, size, and effects of selected text.
Slider/Dial	Configures the appearance of the slider (for a horizontal or vertical gauge) and dial (for a circular gauge).
Warning Zones	Configures the color and percentage for each warning zone.

Note: Specific parameters are discussed in *Detailed Procedures*.

Procedure Overview

Table 15-2: GWXGauge ActiveX Control

To Do This	Follow These Steps:
Insert a GWXGauge ActiveX Control	In M-Graphics, on the Edit menu, select Insert New Object. Select Create Control and ICONICS GWXGauge ActiveX for the object type. Click OK.
Edit General Parameters	Double-click on the GWXGauge control. Select the General tab. Modify the parameters.
Edit Background Parameters	Double-click on the GWXGauge control. Select the Background tab. Modify the parameters.
Edit Caption Parameters	Double-click on the GWXGauge control. Select the Caption tab. Modify the parameters.
Edit Bar Parameters	Double-click on the GWXGauge control. Select the Bar tab. Modify the parameters.
Edit Needle Parameters	Double-click on the GWXGauge control. Select the Needle tab. Modify the parameters.
Edit Value Parameters	Double-click on the GWXGauge control. Select the Value tab. Modify the parameters.
Edit Scale1 Parameters	Double-click on the GWXGauge control. Select the Scale1 tab. Modify the parameters.
Edit Scale2 Parameters	Double-click on the GWXGauge control. Select the Scale2 tab. Modify the parameters.
Edit Fonts Parameters	Double-click on the GWXGauge control. Select the Fonts tab. Modify the parameters.
Edit Slider/Dial Parameters	Double-click on the GWXGauge control. Select the Slider/Dial tab. Modify the parameters.
Edit Warning Zones Parameters	Double-click on the GWXGauge control. Select the Warning Zones tab. Modify the parameters.
View About Information	Double-click on the GWXGauge control. Select the General tab. Select the About button.

Detailed Procedures

Inserting a GWXGauge ActiveX Control

To insert a GWXGauge ActiveX control:

1. In M-Graphics on the Edit menu, select Insert New Object. The Insert Object dialog box appears (Figure 15-2).

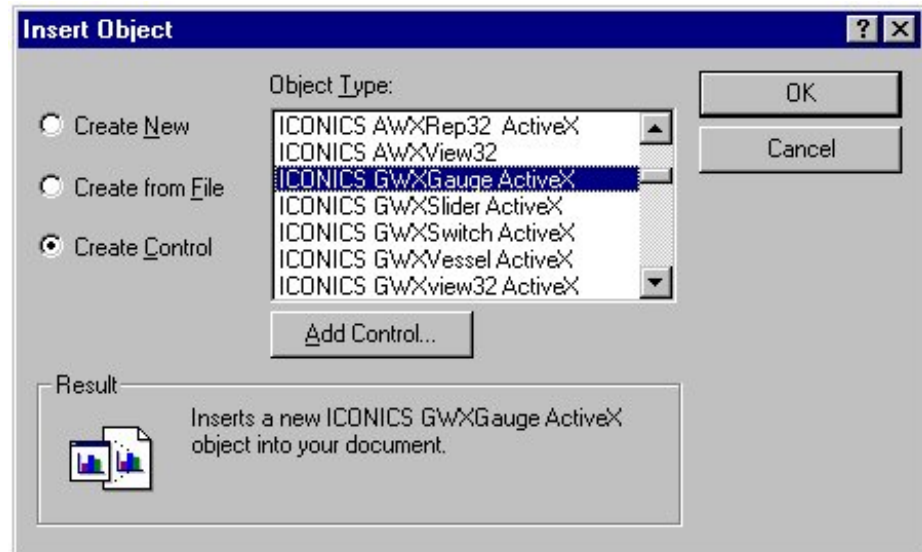



Figure 15-2: Insert Object Dialog Box

2. Select Create Control and ICONICS GWXGauge ActiveX for the object type.
3. Click OK. A GWXGauge ActiveX control is added to the display (Figure 15-3).

Note: As a shortcut, insert a GWXGauge ActiveX control by clicking on the  icon in the M-Graphics ActiveX toolbar.

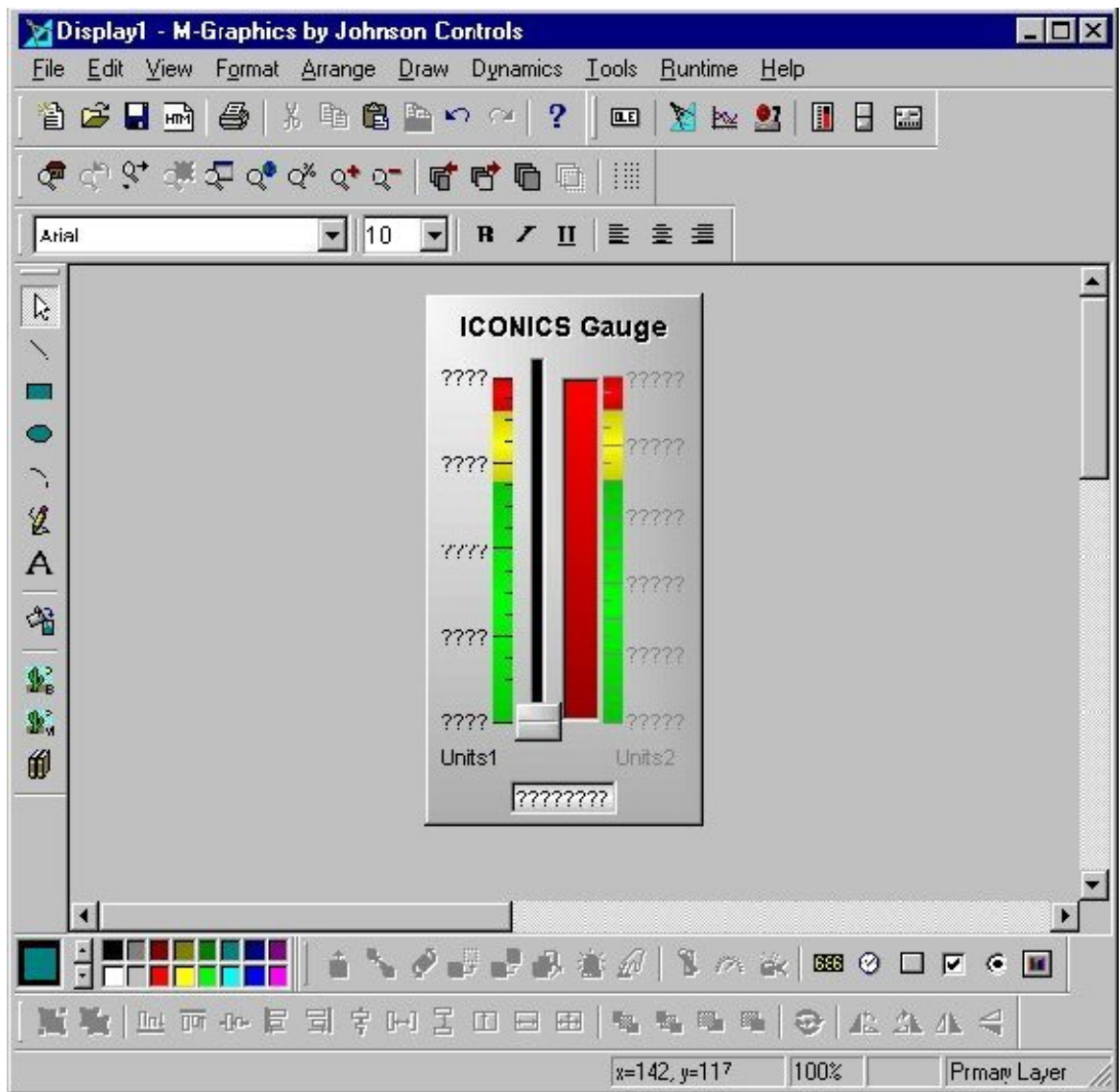


Figure 15-3: GWXGauge ActiveX in M-Graphics Display

Editing General Parameters

To edit general parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears (Figure 15-4).

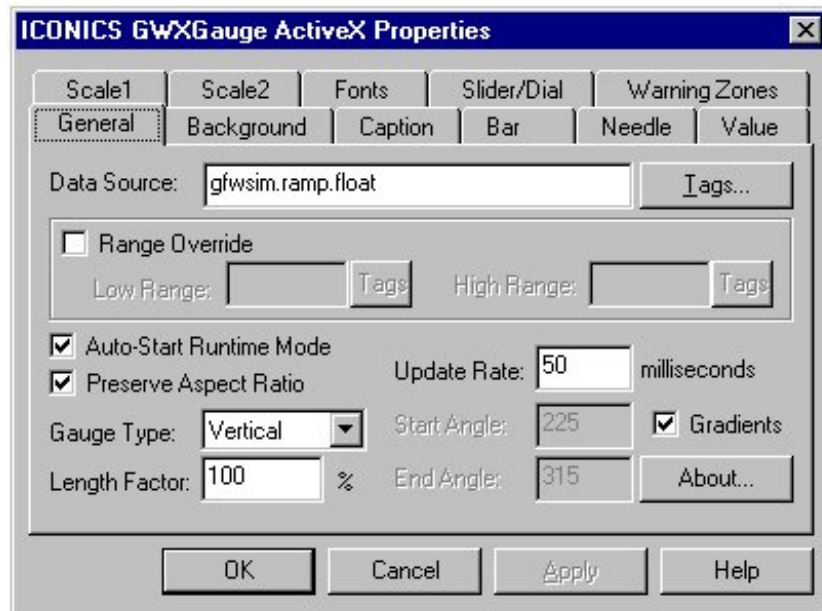


Figure 15-4: GWXGauge ActiveX Properties Dialog Box: General Tab

2. Select the General tab.
3. Modify the parameters using Table 15-3.

Note: Uncheck Preserve Aspect Ratio when changing the Gauge Type from the default Vertical to a Horizontal or Circular gauge type.

Table 15-3: General Tab Parameters

Parameter	Description
Data Source	Displays the current data source.
Tags	Allows user to browse and select a tag.
Range Override	Allows user to override the ranges associated with the selected tag.
Auto-Start Runtime Mode	If checked, automatically starts processing data when display is put in Runtime mode.
Preserve Aspect Ratio	If checked, maintains gauge ratio when the user changes its dimensions.
Update Rate	Determines how fast the system updates the data being displayed through the gauge in milliseconds.
Gauge Type	Displays the available gauge types in a drop-down menu: vertical, horizontal, or circular.
Length Factor	Determines the diameter (circular) or length (vertical or horizontal) and the length of the dial hand for the gauge.
Start Angle	Determines where in the display the gauge begins; for circular gauges only.
End Angle	Determines where in the display the gauge ends; for circular gauges only.
Gradients	Determines if the gauge appears to be light-sourced.
About	Displays the About information.

Editing Background Parameters

To edit background parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears.
2. Select the Background tab (Figure 15-5).

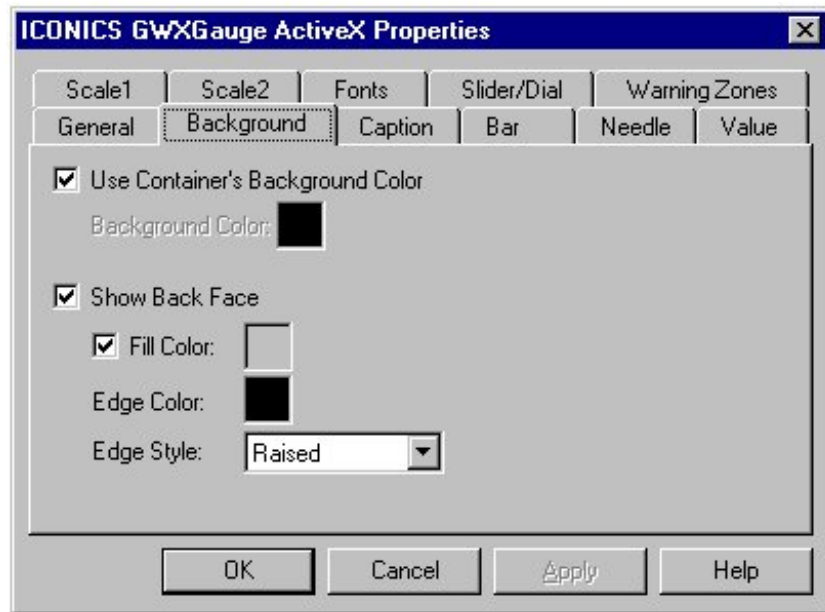


Figure 15-5: GWXGauge ActiveX Properties Dialog Box: Background Tab

3. Modify the parameters using Table 15-4.

Table 15-4: Background Tab Parameters

Parameter	Description
Use Container's Background Color	If checked, the background color of the gauge matches the background color of the container. If unchecked, allows user to select a color for the background of the gauge, which has a border around it.
Show Back Face	If checked, the gauge has an edge border.
Fill Color	Allows user to select a color for the gauge. Default is gray.
Edge Color	Allows user to select a color for the edge of the gauge.
Edge Style	Allows user to select a style for the edge of the gauge: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge. Default is raised.

Editing Caption Parameters

To edit caption parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears.
2. Select the Caption tab (Figure 15-6).

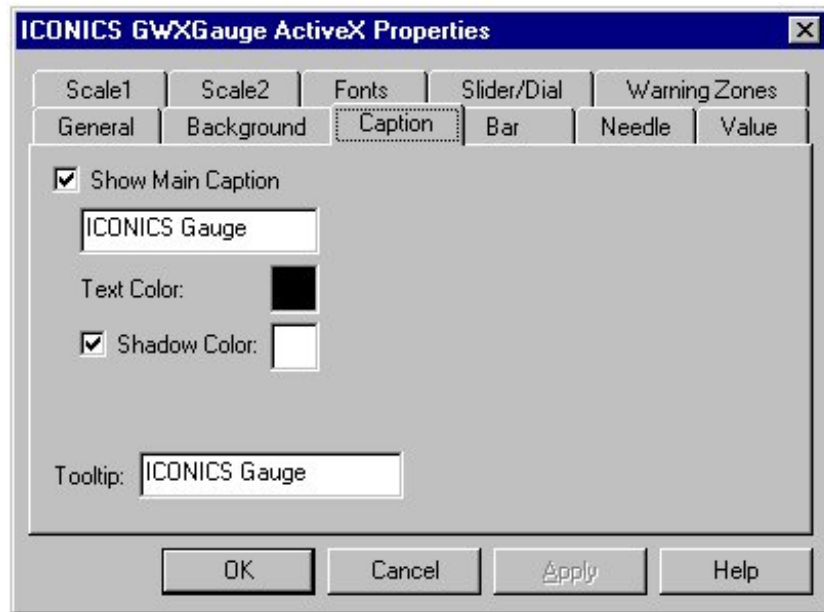


Figure 15-6: GWXGauge ActiveX Properties Dialog Box: Caption Tab

3. Modify the parameters using Table 15-5.

Table 15-5: Caption Tab Parameters

Parameter	Description
Show Main Caption	If checked, the text entered underneath appears as the title of the gauge.
Text Color	Allows user to select a color for the text of the gauge.
Shadow Color	If checked, allows user to select a color for the shadow of the caption text.
ToolTip	Allows user to enter a word, phrase, or message to appear as a ToolTip to the user.

Editing Bar Parameters

Note: The bar parameters apply only to a horizontal or vertical gauge.

To edit bar parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears.
2. Select the Bar tab (Figure 15-7).

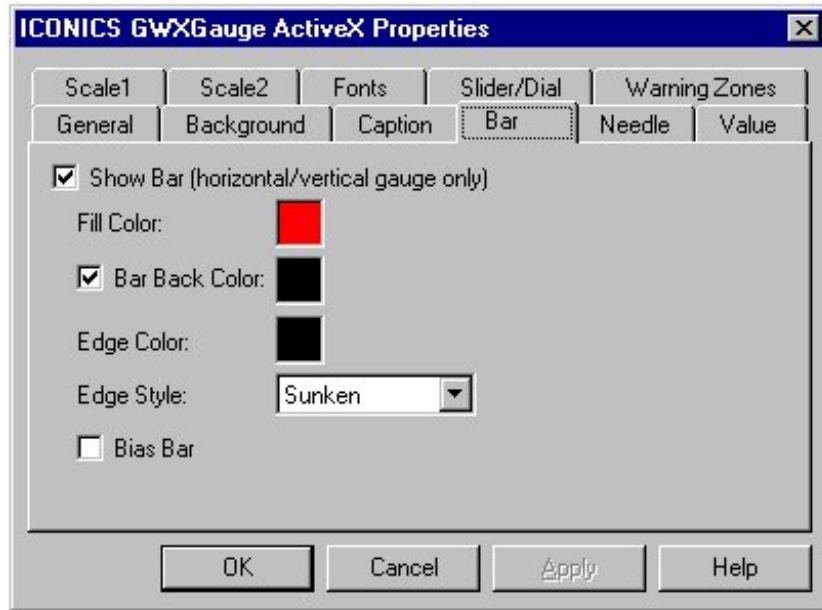


Figure 15-7:GWXGauge ActiveX Properties Dialog Box: Bar Tab

3. Modify the parameters using Table 15-6.

Table 15-6: Bar Tab Parameters

Parameter	Description
Show Bar	If checked, a bar appears next to the slider.
Fill Color	Allows user to select or create a color for the bar of the gauge.
Bar Back Color	Determines the color behind the resizing bar.
Edge Color	Determines the color of the edge of the bar.
Edge Style	Determines the style of the edge of the bar: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge.
Bias Bar	If checked, changes the configuration of the bar so that the bar resizes from its center.

Editing Needle Parameters

Note: The needle parameters apply only to a circular gauge.

To edit needle parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears.
2. Select the Needle tab (Figure 15-8).

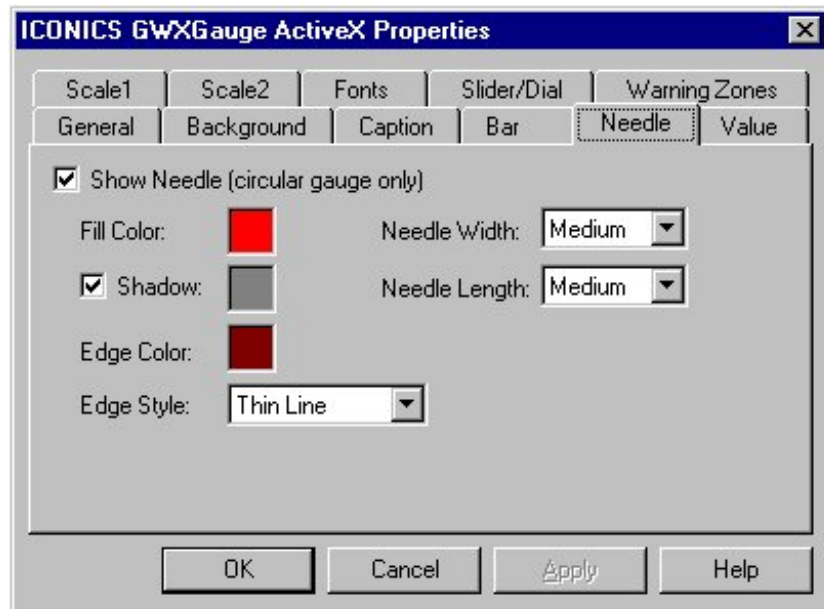


Figure 15-8: GWXGauge ActiveX Properties Dialog Box: Needle Tab

3. Modify the parameters using Table 15-7.

Table 15-7: Needle Tab Parameters

Parameter	Description
Show Needle	If checked, the gauge has a needle.
Fill Color	Determines the color of the needle.
Shadow	Determines the color of the shadow of the needle.
Edge Color	Determines the color of the edge of the needle.
Edge Style	Determines the style of the edge of the needle: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge.
Needle Width	Configures the width of the needle: small, medium, or large.
Needle Length	Configures the length of the needle: small, medium, or large.

Editing Value Parameters

To edit value parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears.
2. Select the Value tab (Figure 15-9).

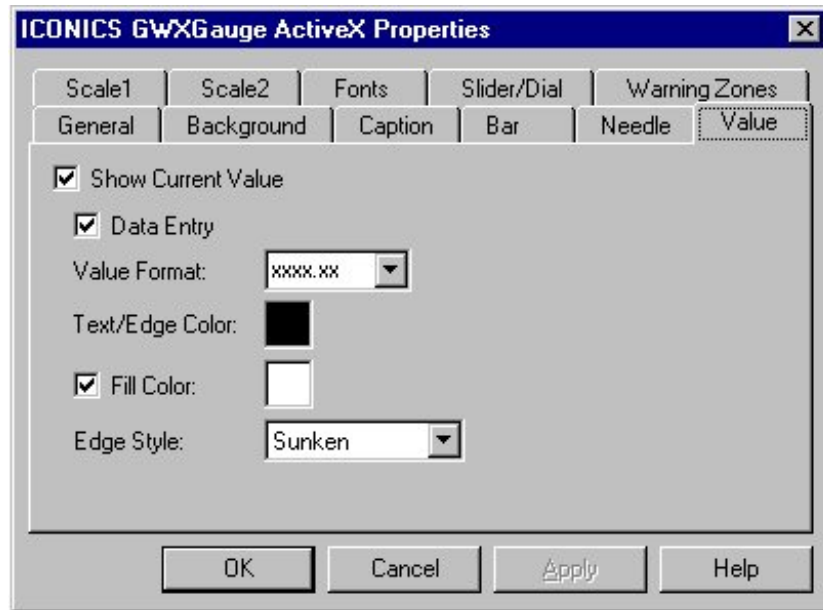


Figure 15-9: GWXGauge ActiveX Properties Dialog Box: Value Tab

3. Modify the parameters using Table 15-8.

Table 15-8: Value Tab Parameters

Parameter	Description
Show Current Value	If checked, the current value of the tag is displayed in a box at the bottom of the gauge.
Data Entry	If checked, user may type in new data values into the box when in Runtime mode.
Value Format	Allows user to determine how the data is displayed from 21 numeric formats such as xxxx.
Text/Edge Color	Determines the color of the text.
Fill Color	Determines the color of the box that displays the values.
Edge Style	Determines the style of the edge of the box.

Editing Scale1 Parameters

To edit scale1 parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears.
2. Select the Scale1 tab (Figure 15-10).

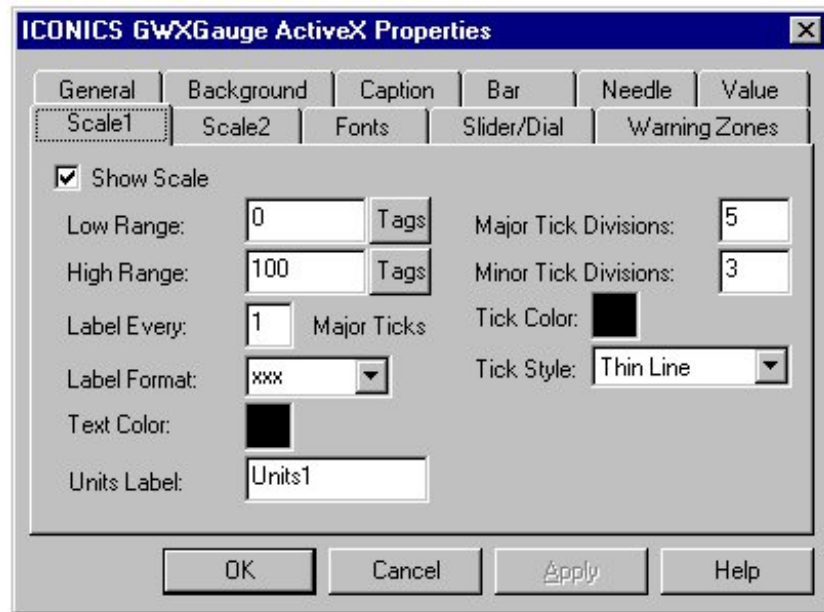


Figure 15-10: GWXGauge ActiveX Properties Dialog Box: Scale1 Tab

3. Modify the parameters using Table 15-9.

Table 15-9: Scale1 Tab Parameters

Parameter	Description
Show Scale	If checked, the gauge has a scale.
Low/High Range	Determines the low and high numbers for the scale.
Label Every	Determines the number of major ticks between labels.
Label Format	Allows user to determine how the label is displayed from 21 formats such as xxxx.
Text Color	Determines the color of the text.
Units Label	Determines the unit text for the labels.
Major Tick Divisions	Determines the number of ticks between major divisions.
Minor Tick Divisions	Determines the number of ticks between minor divisions.
Tick Color	Determines the color of the ticks.
Tick Style	Determines the style of the ticks: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge.

Editing Scale2 Parameters

To edit scale2 parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears.
2. Select the Scale2 tab (Figure 15-11).

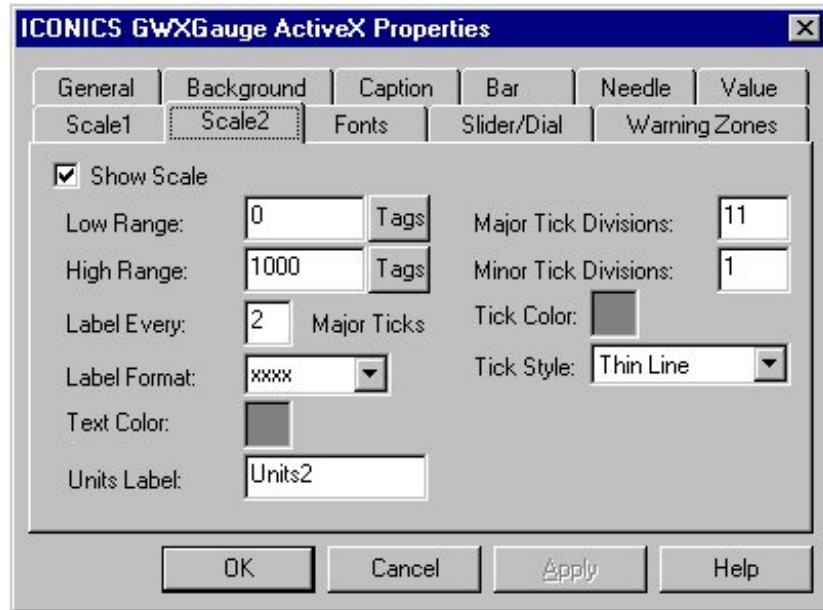


Figure 15-11: GWXGauge ActiveX Properties Dialog Box: Scale2 Tab

3. Modify the parameters using Table 15-10.

Table 15-10: Scale2 Tab Parameters

Parameter	Description
Show Scale	If checked, the gauge has a scale.
Low/High Range	Determines the low and high numbers for the scale.
Label Every	Determines the number of major ticks between labels.
Label Format	Allows user to determine how the label is displayed from 21 formats such as xxxx.
Text Color	Determines the color of the text.
Units Label	Determines the unit text for the labels.
Major Tick Divisions	Determines the number of ticks between major divisions.
Minor Tick Divisions	Determines the number of ticks between minor divisions.
Tick Color	Determines the color of the ticks.
Tick Style	Determines the style of the ticks: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge.

Editing Fonts Parameters

To edit fonts parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears.
2. Select the Fonts tab (Figure 15-12).

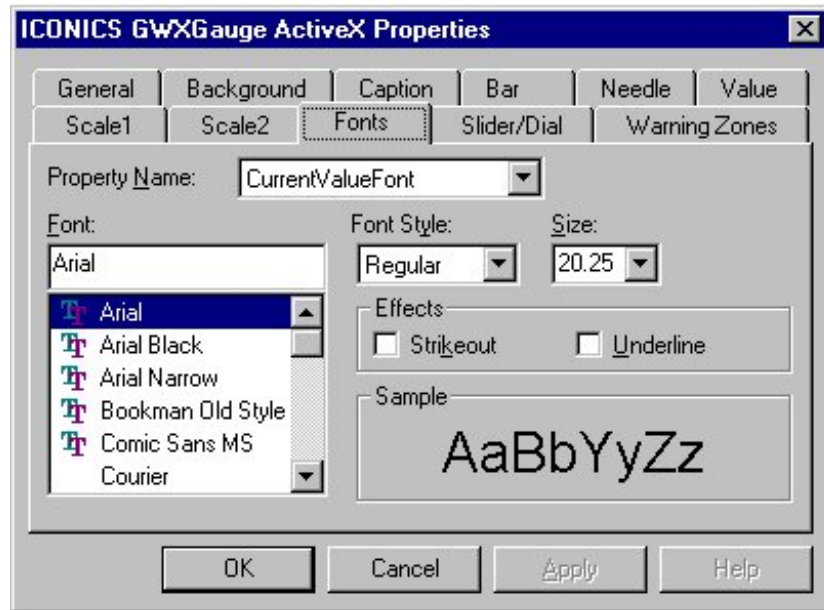


Figure 15-12: GWXGauge ActiveX Properties Dialog Box: Fonts Tab

3. Modify the parameters using Table 15-11.

Table 15-11: Fonts Tab Parameters

Parameter	Description
Property Name	Allows user to choose which text to change the font of: CurrentValueFont, MainCaptionFont, ScaleOneLabelFont, or ScaleTwoLabelFont.
Font	Displays the current font and a scrolling list of fonts available.
Font Style	Determines the style of the font: regular, bold, italic, or bold italic.
Size	Allows user to select the size of the font from a drop-down menu.
Effects	Allows strikeouts and underlining of the text chosen.
Sample	Displays a real-size example of the font, style, and size chosen.

Editing Slider/Dial Parameters

Note: The slider track applies to only a horizontal or vertical gauge. The knob section applies to all types of gauges.

To edit slider/dial parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears.
2. Select the Slider/Dial tab (Figure 15-13).

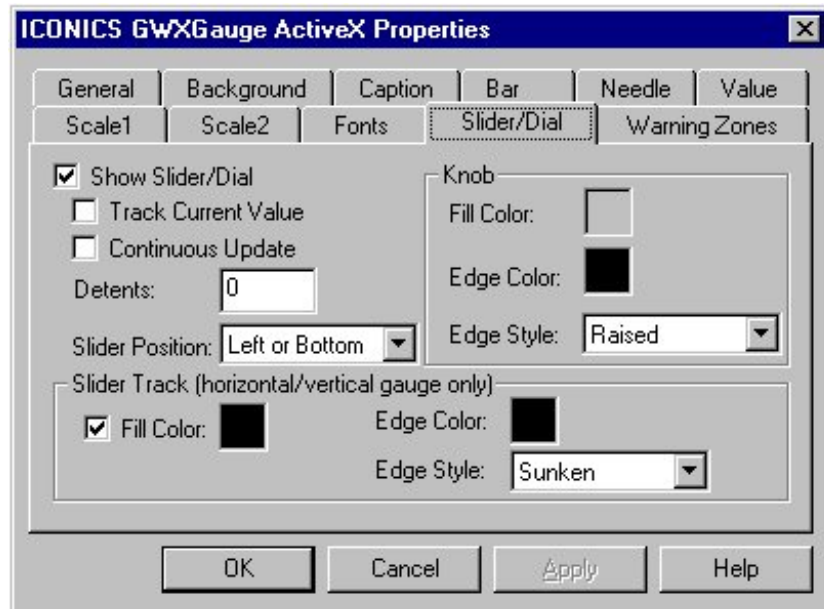


Figure 15-13: GWXGauge ActiveX Properties Dialog Box: Slider/Dial Tab

3. Modify the parameters using Table 15-12.

Table 15-12: Slider/Dial Tab Parameters

Parameters		Description
Show Slider/Dial		If checked, the gauge has a slider/dial.
Track Current Value		If checked, the current value is tracked.
Continuous Update		If checked, the data is updated continuously.
Detents		Allows user to set the increments that the slider/dial will move within the gauge.
Slider Position		Determines where the slider will be placed: left or bottom or right or top.
Slider Track	Fill Color	Determines the color of the bar the slider moves on.
	Edge Color	Determines the color of the edge of the slider track.
	Edge Style	Determines the style of the edge of the slider track: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge.
Knob	Fill Color	Determines the color of the dial or slider.
	Edge Color	Determines the edge color of the dial or slider.
	Edge Style	Determines the style of the edge of the dial or slider: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge.

Editing Warning Zones Parameters

To edit warning zones parameters:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears.
2. Select the Warning Zones tab (Figure 15-14).

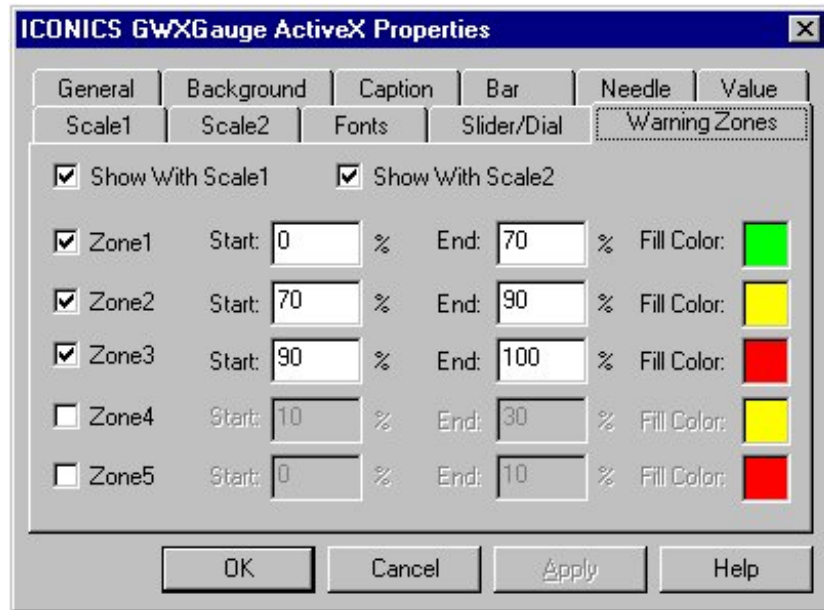


Figure 15-14: GWXGauge ActiveX Properties Dialog Box: Warning Zones Tab

3. Modify the parameters using Table 15-13.

Table 15-13: Warning Zones Tab Parameters

Parameter	Description
Show With Scale1	If checked, the selected warning zones display on scale1.
Show With Scale2	If checked, the selected warning zones display on scale2.
Zone1, Zone2, etc.	Allows user to configure up to five zones for warnings on the gauge.
Start	Allows user to define the start value of the zone in percentages.
End	Allows user to define the end value of the zone in percentages.
Fill Color	Determines the color of the warning for the selected zone.

Viewing the About Information

To view the about information:

1. Double-click on the GWXGauge ActiveX control. The GWXGauge ActiveX Properties dialog box appears (Figure 15-15).

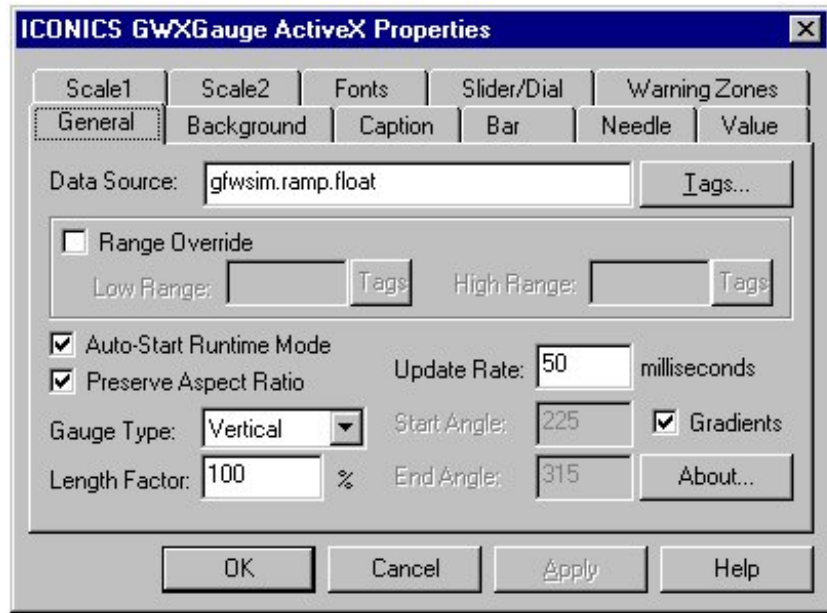


Figure 15-15: GWXGauge ActiveX Properties Dialog Box: General Tab

2. Select the General tab.
3. Select the About button. The Iconics About Box appears (Figure 15-16).

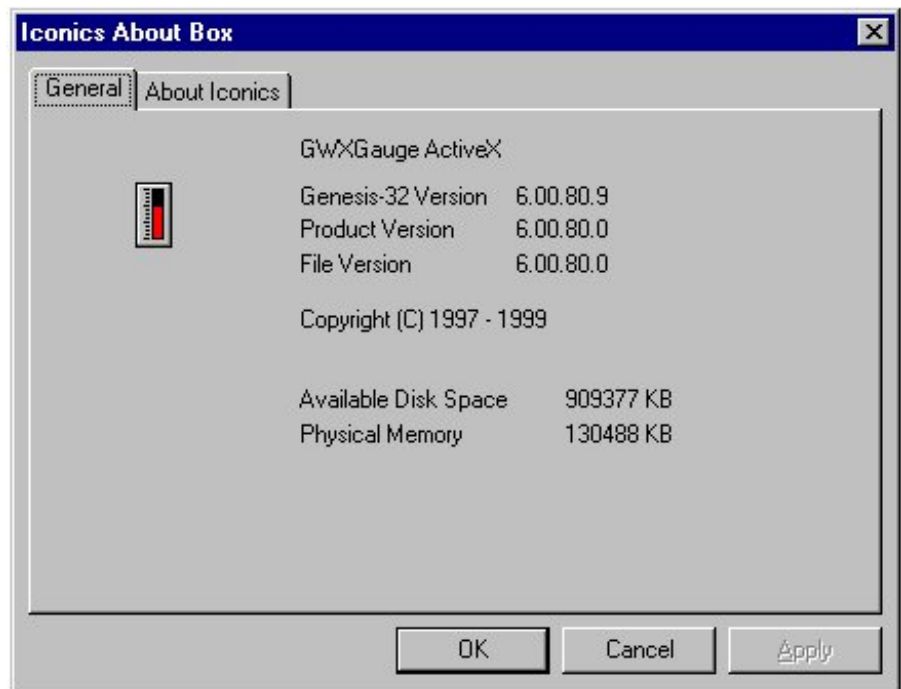


Figure 15-16: Iconics About Box

4. Select the General tab. The information listed includes version, copyright, etc.

GWXSwitch ActiveX Control

Introduction

The GWXSwitch is an ActiveX control that can be inserted into the M-Graphics application to display and modify binary output. This chapter describes how to:

- insert a GWXSwitch ActiveX control
- edit general parameters
- edit background parameters
- edit caption parameters
- edit labels parameters
- edit fonts parameters
- view about information

Key Concepts

GWXSwitch ActiveX Control

Insert the GWXSwitch ActiveX control into the M-Graphics application. The GWXSwitch ActiveX displays and modifies data in a virtual instrument panel format on graphics. Use the GWXSwitch ActiveX to output binary changes, such as start and stop, to field devices. The GWXSwitch Active X supports connection to OPC compliant servers and provides excellent capabilities for managing data.

GWXSwitch ActiveX Properties

The GWXSwitch ActiveX Properties dialog box is the configuration interface to the GWXSwitch ActiveX. Double-clicking anywhere in the GWXSwitch ActiveX client during Configure mode opens the GWXSwitch ActiveX Properties dialog box (Figure 16-1). The dialog box fields are described in Table 16-1.

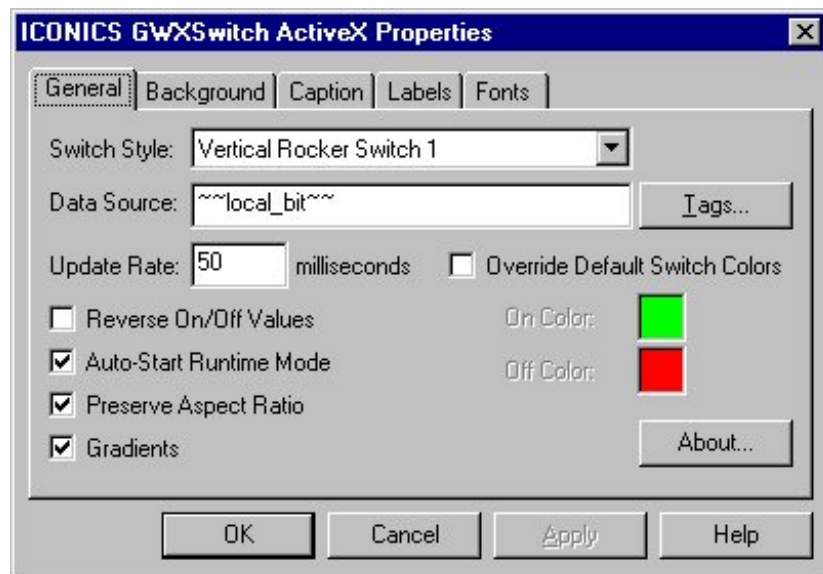


Figure 16-1: GWXSwitch ActiveX Properties Dialog Box

Table 16-1: GWXSwitch ActiveX Properties Dialog Box

Properties Page	Description
General	Allows user to select the tag used as the data source for the switch.
Background	Configures the appearance of the switch within a container.
Caption	Configures the caption for the switch and the text for the ToolTip.
Labels	Allows user to configure labels for the switch.
Fonts	Determines the font, font style, size, and effects of selected text.

Note: Specific parameters are discussed in *Detailed Procedures*.

Procedure Overview

Table 16-2: GWXSwitch ActiveX Control

To Do This	Follow These Steps:
Insert a GWXSwitch ActiveX Control	In M-Graphics, on the Edit menu, select Insert New Object. Select Create Control and ICONICS GWXSwitch ActiveX for the object type. Click OK.
Edit General Parameters	Double-click on the GWXSwitch control. Select the General tab. Modify the parameters.
Edit Background Parameters	Double-click on the GWXSwitch control. Select the Background tab. Modify the parameters.
Edit Caption Parameters	Double-click on the GWXSwitch control. Select the Caption tab. Modify the parameters.
Edit Labels Parameters	Double-click on the GWXSwitch control. Select the Labels tab. Modify the parameters.
Edit Fonts Parameters	Double-click on the GWXSwitch control. Select the Fonts tab. Modify the parameters.
View About Information	Double-click on the GWXSwitch control. Select the General tab. Select the About button.

Detailed Procedures

Inserting a GWXSwitch ActiveX Control

To insert a GWXSwitch ActiveX control:

1. In M-Graphics on the Edit menu, select Insert New Object. The Insert Object dialog box appears (Figure 16-2).

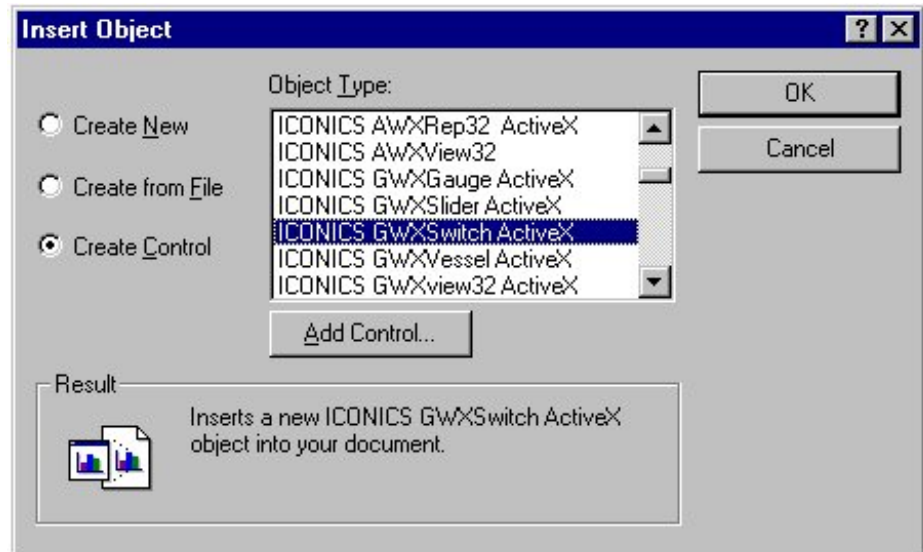



Figure 16-2: Insert Object Dialog Box

2. Select Create Control and ICONICS GWXSwitch ActiveX for the object type.
3. Click OK. A GWXSwitch ActiveX control is added to the display (Figure 16-3).

Note: As a shortcut, insert a GWXSwitch ActiveX control by clicking on the  icon in the M-Graphics ActiveX toolbar.

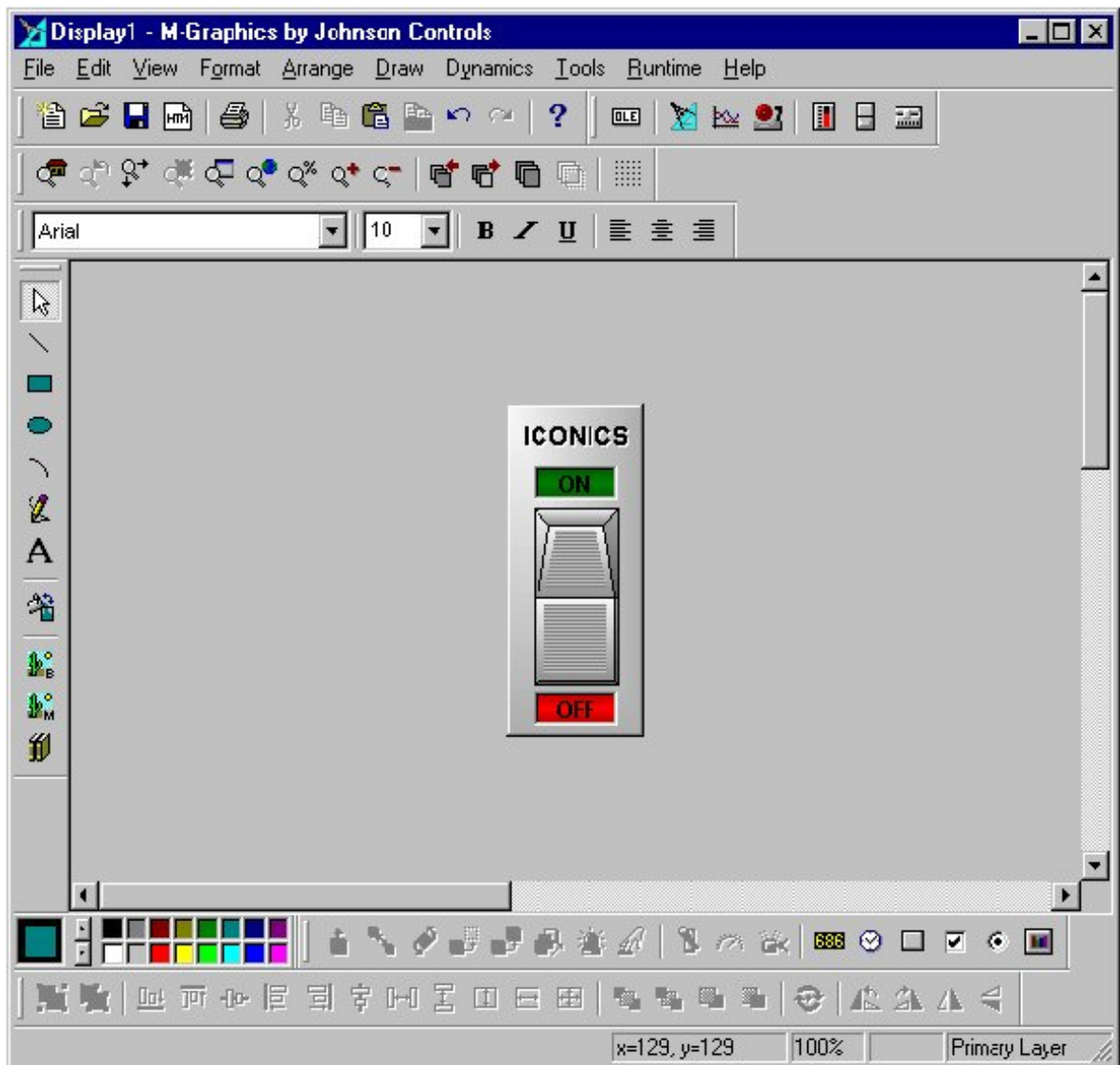


Figure 16-3: GWXSwitch ActiveX in M-Graphics Display

Editing General Parameters

To edit general parameters:

1. Double-click on the GWXSwitch ActiveX control. The GWXSwitch ActiveX Properties dialog box appears (Figure 16-4).

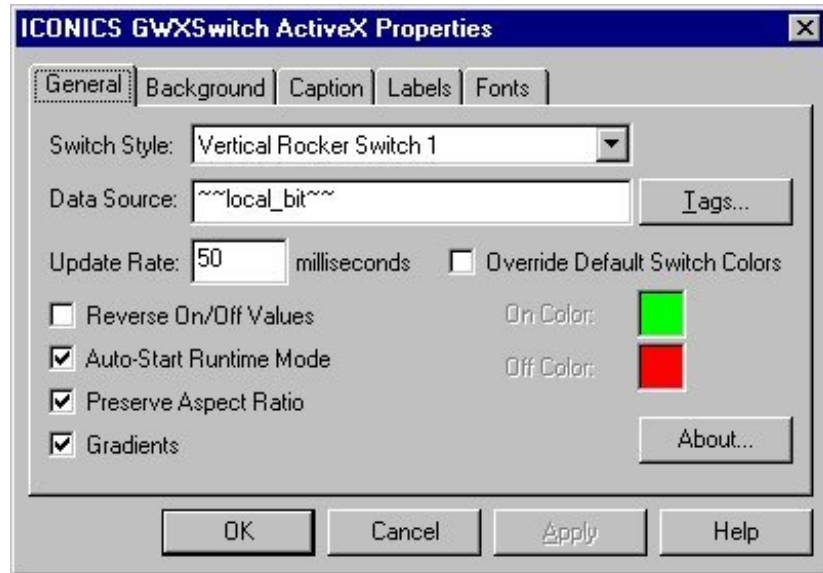


Figure 16-4: GWXSwitch ActiveX Properties Dialog Box: General Tab

2. Select the General tab.
3. Modify the parameters using Table 16-3.

Note: Uncheck Preserve Aspect Ratio when changing the Switch Style from the default Vertical Rocker Switch 1 to another Switch Style.

Table 16-3: General Tab Parameters

Parameter	Description
Switch Style	Allows user to select a switch type: Vertical Rocker, Vertical Slider, Vertical Toggle, Horizontal Rocker, Horizontal Slider, Horizontal Toggle, Dial, or On/Off.
Data Source	Displays the current data source.
Tags	Allows user to browse and select a tag.
Update Rate	Determines how fast the system will update the data being displayed through the switch in milliseconds.
Reverse On/Off Values	Allows the user to reverse the values of the associated tag depending on whether the switch is on or off.
Auto-Start Runtime Mode	If checked, automatically starts processing data when display is put in Runtime mode.
Preserve Aspect Ratio	If checked, maintains switch ratio when the user changes its dimensions.
Gradients	Determines if the switch appears to be light-sourced.
Override Default Switch Colors	If checked, allows user to select colors for the switch instead of using the default colors.
About	Displays the About information.

Editing Background Parameters

To edit background parameters:

1. Double-click on the GWXSwitch ActiveX control. The GWXSwitch ActiveX Properties dialog box appears.
2. Select the Background tab (Figure 16-5).



Figure 16-5: GWXSwitch Active X Properties Dialog Box: Background Tab

3. Modify the parameters using Table 16-4.

Table 16-4: Background Tab Parameters

Parameter	Description
Use Container's Background Color	If checked, the background color of the switch matches the background color of the container. If unchecked, allows user to select a color for the background of the switch, which has a border around it.
Show Back Face	If checked, the switch has an edge border.
Fill Color	Allows user to select a color for the switch. Default is gray.
Edge Style	Allows user to select a style for the edge of the switch: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge.

Editing Caption Parameters

To edit caption parameters:

1. Double-click on the GWXSwitch ActiveX control. The GWXSwitch ActiveX Properties dialog box appears.
2. Select the Caption tab (Figure 16-6).

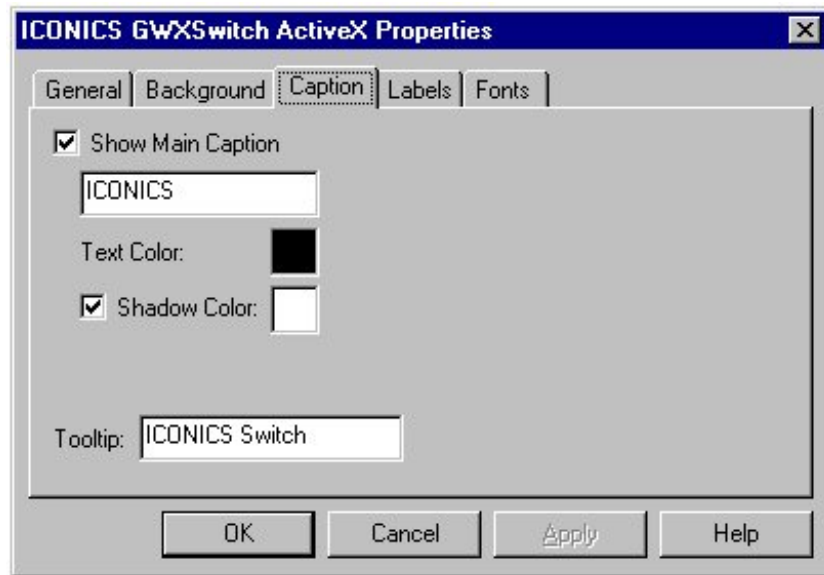


Figure 16-6: GWXSwitch ActiveX Properties Dialog Box: Caption Tab

3. Modify the parameters using Table 16-5.

Table 16-5: Caption Tab Parameters

Parameter	Description
Show Main Caption	If checked, the text entered underneath appears as the title of the switch.
Text Color	Allows user to select a color for the text of the switch.
Shadow Color	If checked, allows the user to select a color for the shadow of the caption text.
ToolTip	Allows user to enter a word, phrase, or message to appear as a ToolTip to the user.

Editing Labels Parameters

To edit labels parameters:

1. Double-click on the GWXSwitch ActiveX control. The GWXSwitch ActiveX Properties dialog box appears.
2. Select the Labels tab (Figure 16-7).



Figure 16-7: GWXSwitch ActiveX Properties Dialog Box: Labels Tab

3. Modify the parameters using Table 16-6.

Table 16-6: Labels Tab Parameters

Parameter	Description
Show Labels	If checked, the switch has labels.
Edge Style	Allows user to select a style for the edge of the labels: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge
"On" Label	Allows user to enter the text for the "on" label.
	Fill Color When On Determines the color of the "on" label when the switch is on.
	Fill Color When Off Determines the color of the "on" label when the switch is off.
	Text Color Determines the color of the text for the "on" label.
"Off" Label	Allows user to enter the text for the "off" label.
	Fill Color When On Determines the color of the "off" label when the switch is on.
	Fill Color When Off Determines the color of the "off" label when the switch is off.
	Text Color Determines the color of the text for the "off" label.

Editing Fonts Parameters

To edit fonts parameters:

1. Double-click on the GWXSwitch ActiveX control. The GWXSwitch ActiveX Properties dialog box appears.
2. Select the Fonts tab (Figure 16-8).

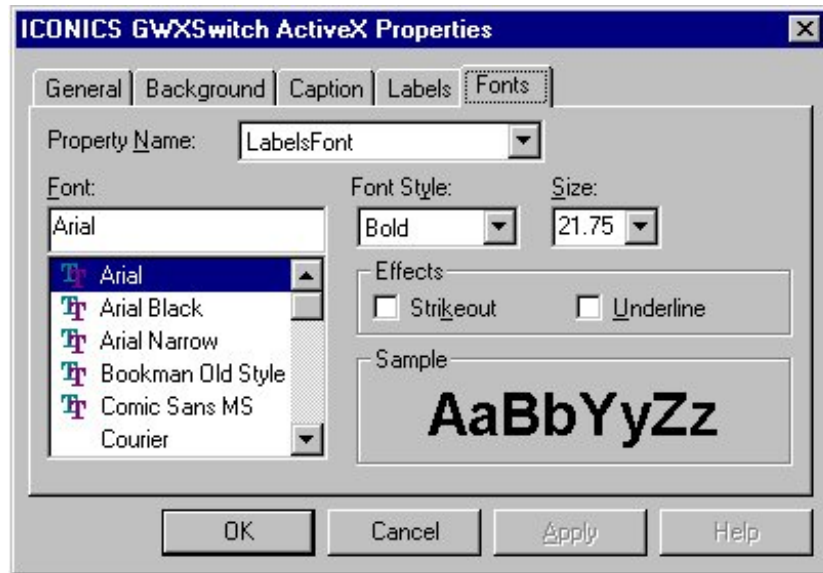


Figure 16-8: GWXSwitch ActiveX Properties Dialog Box: Fonts Tab

3. Modify the parameters using Table 16-7.

Table 16-7: Fonts Tab Parameters

Parameter	Description
Property Name	Allows user to choose which text to change the font of: LabelsFonts or MainCaptionFont.
Font	Displays the current font and a scrolling list of fonts available.
Font Style	Determines the style of the font: regular, bold, italic, or bold italic.
Size	Allows user to select the size of the font from a drop-down menu.
Effects	Allows strikeouts and underlining of the text chosen.
Sample	Displays a real-size example of the font, style, and size chosen.

Viewing the About Information

To view the about information:

1. Double-click on the GWXSwitch ActiveX control. The GWXSwitch ActiveX Properties dialog box appears (Figure 16-9).

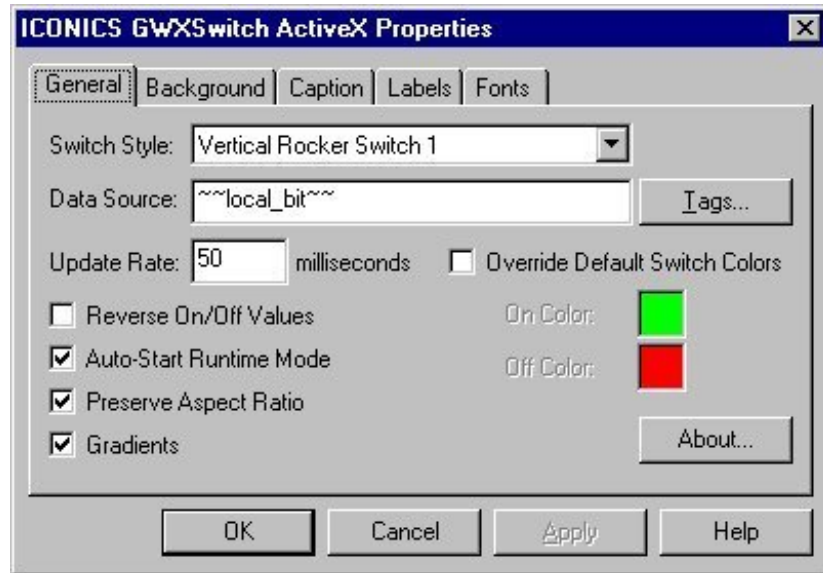


Figure 16-9: GWXSwitch ActiveX Properties Dialog Box: General Tab

2. Select the General tab.
3. Select the About button. The Iconics About Box appears (Figure 16-10).

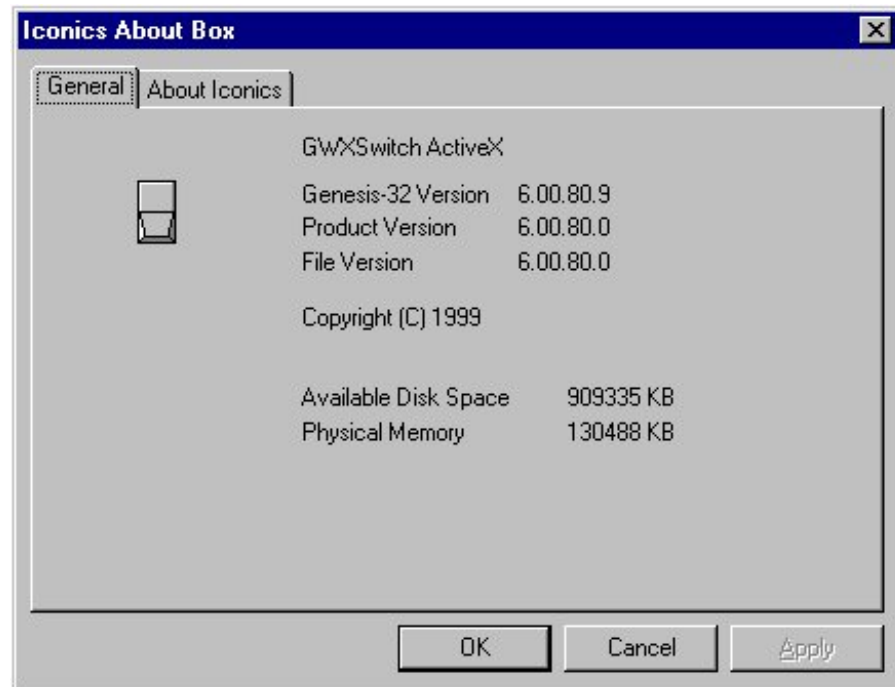


Figure 16-10: Iconics About Box

4. Select the General tab. The information listed includes version, copyright, etc.

GWXSlider ActiveX Control

Introduction

The GWXSlider is an ActiveX control that can be inserted into the M-Graphics application to display and modify analog data. This chapter describes how to:

- insert a GWXSlider ActiveX control
- edit general parameters
- edit slider parameters
- edit slow spin buttons parameters
- edit fast spin buttons parameters
- edit scale parameters
- edit background parameters
- edit caption parameters
- edit value parameters
- edit fonts parameters
- view about information

Key Concepts

GWXSlider ActiveX Control

GWXSlider ActiveX is an ActiveX control that can be inserted in the M-Graphics application. The GWXSlider ActiveX displays and modifies data in a virtual instrument panel format on graphics. The GWXSlider ActiveX supports connection to OPC compliant servers and provides excellent capabilities for managing data.

GWXSlider ActiveX Properties

The GWXSlider ActiveX Properties dialog box is the configuration interface to the GWXSlider ActiveX. Double-clicking anywhere in the GWXSlider ActiveX client during Configure mode opens the GWXSlider ActiveX Properties dialog box (Figure 17-1). The dialog box fields are described in Table 17-1.

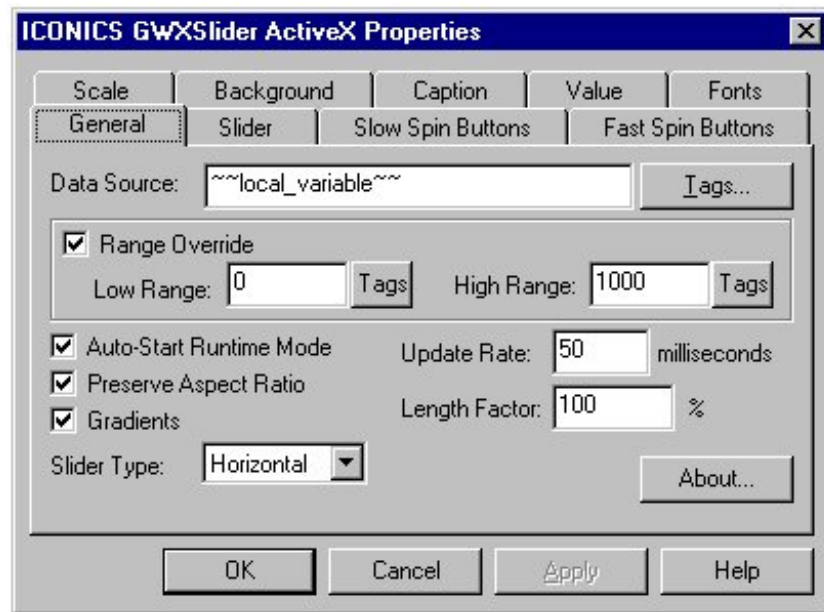


Figure 17-1: GWXSlider ActiveX Properties Dialog Box

Table 17-1: GWXSlider ActiveX Properties Dialog Box

Properties Page	Description
General	Allows user to select the tag used as the data source for the slider.
Slider	Configures the appearance and behavior of the slider knob and track.
Slow Spin Buttons	Configures the appearance and behavior of the slider spin buttons.
Fast Spin Buttons	Configures the appearance and behavior of the slider spin buttons.
Scale	Configures the appearance of the scale of the slider.
Background	Configures the appearance of the slider within the container.
Caption	Configures the caption for the slider and the text for the ToolTip.
Value	Configures the format and style of the text box of the slider.
Fonts	Determines the font, font style, size, effects of selected text.

Note: Specific parameters are discussed in *Detailed Procedures*.

Procedure Overview

Table 17-2: GWXSlider ActiveX Control

To Do This	Follow These Steps:
Insert a GWXSlider ActiveX Control	In M-Graphics, on the Edit menu, select Insert New Object. Select Create Control and ICONICS GWXSlider ActiveX for the object type. Click OK.
Edit General Parameters	Double-click on the GWXSlider control. Select the General tab. Modify the parameters.
Edit Slider Parameters	Double-click on the GWXSlider control. Select the Slider tab. Modify the parameters.
Edit Slow Spin Buttons Parameters	Double-click on the GWXSlider control. Select the Slow Spin Buttons tab. Modify the parameters.
Edit Fast Spin Buttons Parameters	Double-click on the GWXSlider control. Select the Fast Spin Buttons tab. Modify the parameters.
Edit Scale Parameters	Double-click on the GWXSlider control. Select the Scale tab. Modify the parameters.
Edit Background Parameters	Double-click on the GWXSlider control. Select the Background tab. Modify the parameters.
Edit Caption Parameters	Double-click on the GWXSlider control. Select the Caption tab. Modify the parameters.
Edit Value Parameters	Double-click on the GWXSlider control. Select the Value tab. Modify the parameters.
Edit Fonts Parameters	Double-click on the GWXSlider control. Select the Fonts tab. Modify the parameters.
View About Information	Double-click on the GWXSlider control. Select the General tab. Select the About button.

Detailed Procedures

Inserting a GWXSlider ActiveX Control

To insert a GWXSlider ActiveX control:

1. In M-Graphics on the Edit menu, select Insert New Object. The Insert Object dialog box appears (Figure 17-2).

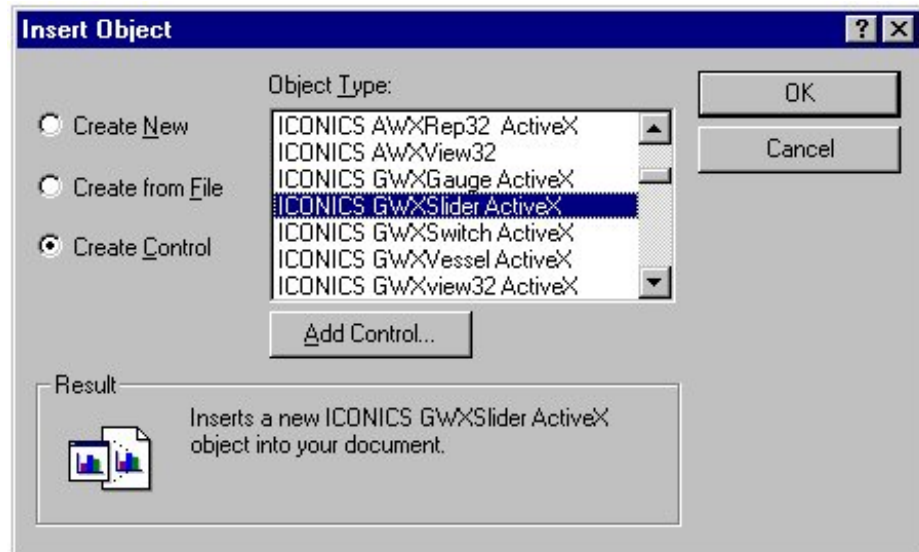



Figure 17-2: Insert Object Dialog Box

2. Select Create Control and ICONICS GWXSlider ActiveX for the object type.
3. Click OK. A GWXSlider ActiveX control is added to the display (Figure 17-3).

Note: As a shortcut, insert a GWXSlider ActiveX control by clicking on the  icon in the M-Graphics ActiveX toolbar.

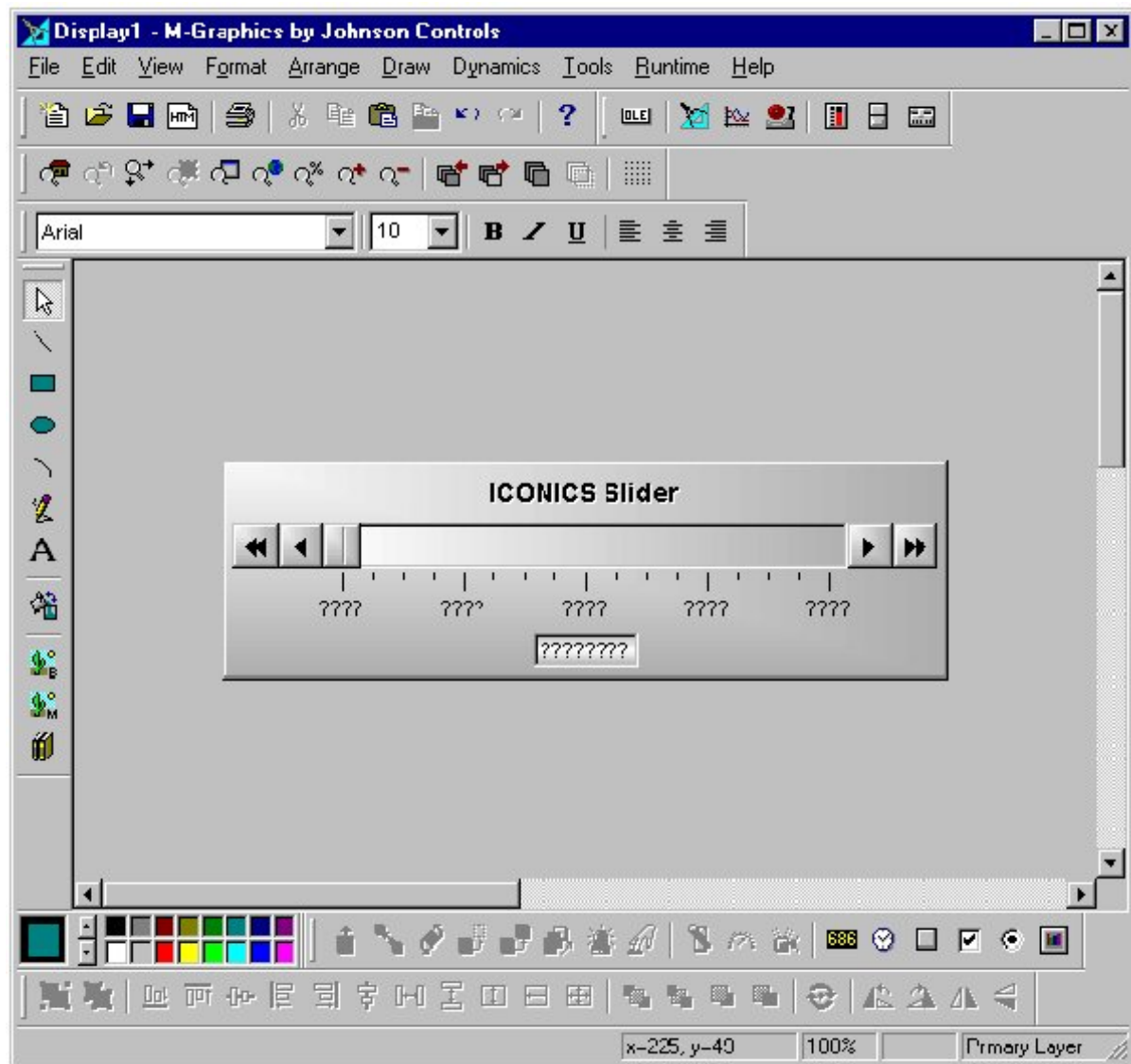


Figure 17-3: GWXSlider ActiveX in M-Graphics Display

Editing General Parameters

To edit general parameters:

1. Double-click on the GWXSlider ActiveX control. The GWXSlider ActiveX Properties dialog box appears (Figure 17-4).

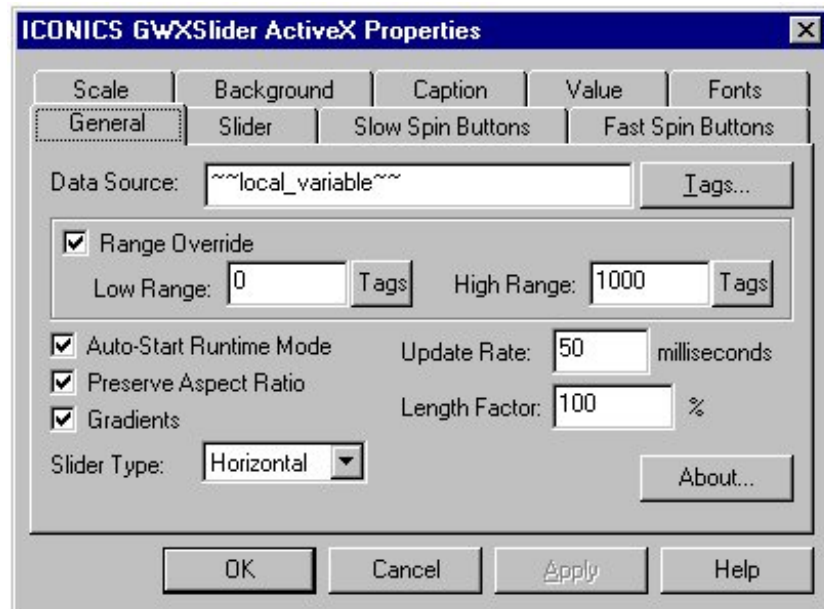


Figure 17-4: GWXSlider ActiveX Properties Dialog Box: General Tab

2. Select the General tab.
3. Modify the parameters using Table 17-3.

Note: Uncheck Preserve Aspect Ratio when changing the Slider Type from the default Horizontal to a Vertical Slider Type.

Table 17-3: General Tab Parameters

Parameter	Description
Data Source	Displays the current data source.
Tags	Allows user to browse and select a tag.
Range Override	Allows user to override the ranges associated with the selected tag.
Auto-Start Runtime Mode	If checked, automatically starts processing data when display is put in Runtime mode.
Preserve Aspect Ratio	If checked, maintains slider ratio when the user changes its dimensions.
Gradients	Determines if the slider appears to be light-sourced.
Slider Type	Displays the available slider types in a drop-down menu: vertical or horizontal.
Update Rate	Determines how fast the system updates the data being displayed through the slider in milliseconds.
Length Factor	Determines the length (vertical or horizontal) of the slider.
About	Displays the About information.

Editing Slider Parameters

To edit slider parameters:

1. Double-click on the GWXSlider ActiveX control. The GWXSlider ActiveX Properties dialog box appears.
2. Select the Slider tab (Figure 17-5).

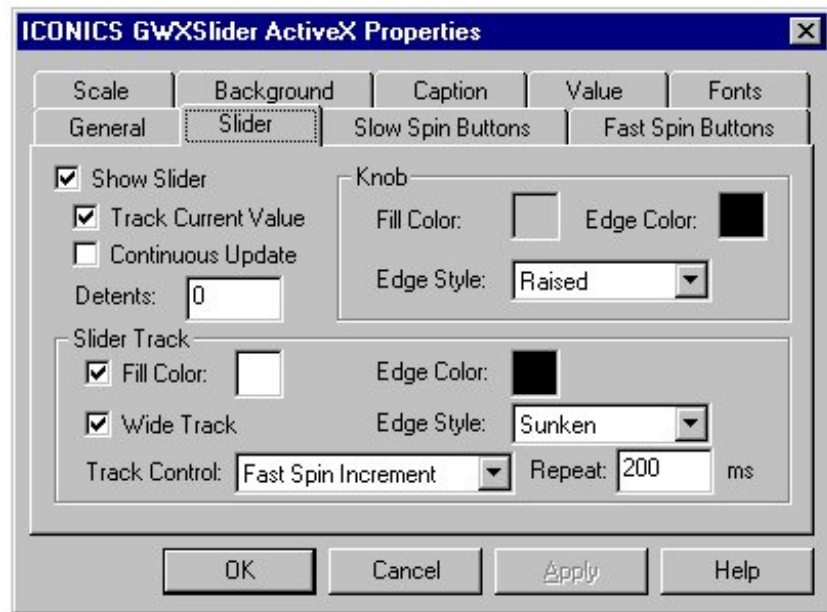


Figure 17-5: GWXSlider ActiveX Properties Dialog Box: Slider Tab

3. Modify the parameters using Table 17-4.

Table 17-4: Slider Tab Parameters

Parameters		Description
Show Slider		If checked, the slider is visible.
Track Current Value		If checked, the current value is tracked.
Continuous Update		If checked, the data is updated continuously.
Detents		Allows user to set the increments that the slider knob will move.
Slider Track	Fill Color	Determines the color of the bar the slider moves on.
	Wide Track	Displays a wide slider track.
	Edge Color	Determines the color of the edge of the slider track.
	Edge Style	Determines the style of the edge of the slider track: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge.
	Track Control	Determines the behavior when user clicks on the slider track: move to next detent, move to detent nearest the mouse pointer, move to mouse pointer position, move by fast spin increment, move by slow spin increment, or no action.
	Repeat	Determines how often the track control action is repeated when the user holds down the mouse button.
Knob	Fill Color	Determines the color of the slider knob.
	Edge Color	Determines the color of the edge of the slider knob.
	Edge Style	Determines the style of the edge of the slider knob: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge.

Editing Slow Spin Buttons Parameters

To edit slow spin buttons parameters:

1. Double-click on the GWXSlider ActiveX control. The GWXSlider ActiveX Properties dialog box appears.
2. Select Slow Spin Buttons tab (Figure 17-6).

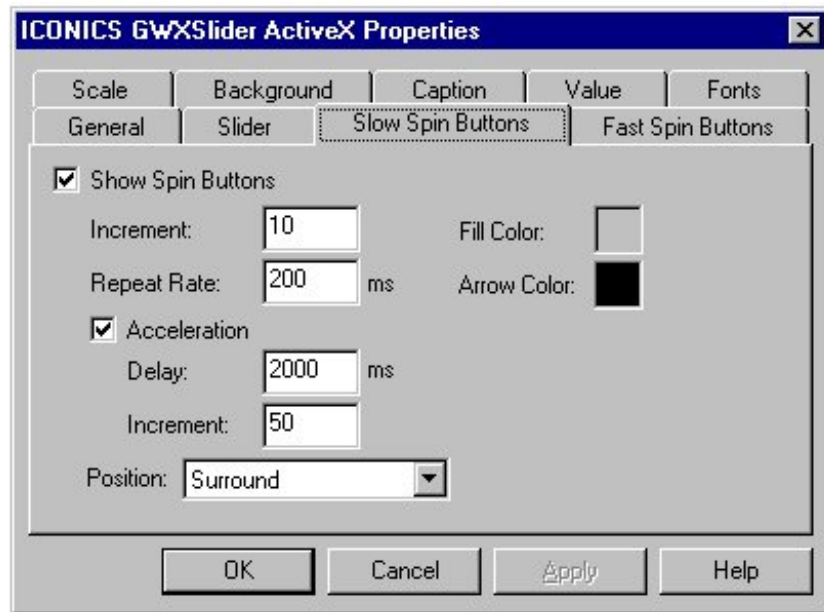


Figure 17-6: GWXSlider ActiveX Properties Dialog Box: Slow Spin Buttons Tab

3. Modify the parameters using Table 17-5.

Table 17-5: Slow Spin Buttons Tab Parameters

Parameter	Description
Show Spin Buttons	If checked, the slider has spin buttons.
Increment	Determines the increment or decrement value when clicking on a spin button.
Repeat Rate	Determines how often the increment action repeats when the user holds down the mouse on a spin button.
Fill Color	Determines the color of the spin buttons.
Arrow Color	Determines the color of the arrow.
Acceleration	Allows user to keep holding down a spin button to accelerate the increment.
Delay	Determines when the acceleration begins after spin buttons are pressed.
Increment	Determines the increment or decrement value for the acceleration option.
Position	Determines the position of the spin buttons relative to the slider: group left or bottom, group right or top, or surround.

Editing Fast Spin Buttons Parameters

To edit fast spin buttons parameters:

1. Double-click on the GWXSlider ActiveX control. The GWXSlider ActiveX Properties dialog box appears.
2. Select Fast Spin Buttons tab (Figure 17-7).

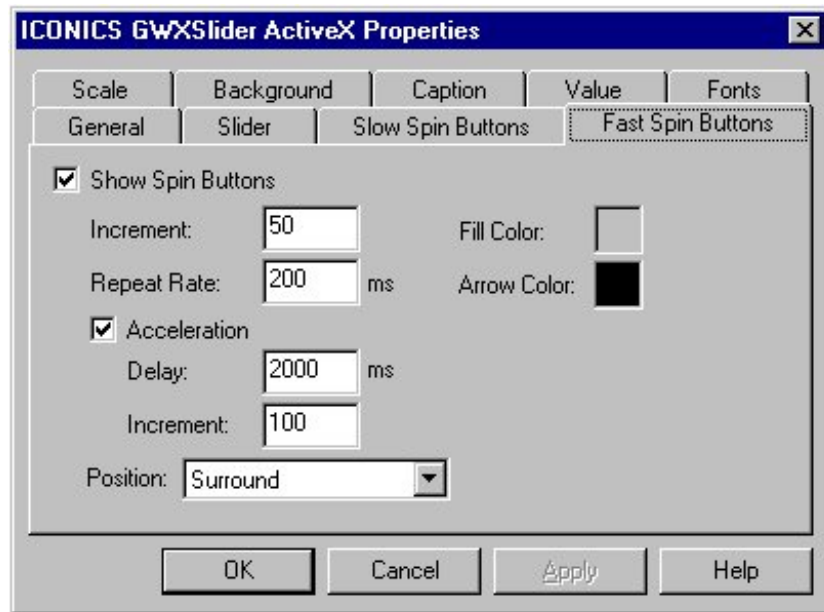


Figure 17-7: GWXSlider ActiveX Properties Dialog Box: Fast Spin Buttons

3. Modify the parameters using Table 17-6.

Table 17-6: Fast Spin Buttons Tab Parameters

Parameter	Description
Show Spin Buttons	If checked, the slider has spin buttons.
Increment	Determines the increment or decrement value when clicking on a spin button.
Repeat Rate	Determines how often the increment action repeats when the user holds down the mouse on a spin button.
Fill Color	Determines the color of the spin buttons.
Arrow Color	Determines the color of the arrow.
Acceleration	Allows user to keep holding down a spin button to accelerate the increment.
Delay	Determines when the acceleration begins after spin buttons are pressed.
Increment	Determines the increment or decrement value for the acceleration option.
Position	Determines the position of the spin buttons relative to the slider: group left or bottom, group right or top, or surround.

Editing Scale Parameters

To edit scale parameters:

1. Double-click on the GWXSlider ActiveX control. The GWXSlider ActiveX Properties dialog box appears.
2. Select the Scale tab (Figure 17-8).



Figure 17-8: GWXSlider ActiveX Properties Dialog Box: Scale Tab

3. Modify the parameters using Table 17-7.

Table 17-7: Scale Tab Parameters

Parameter	Description
Show Scale	If checked, the slider has a scale.
Low Range	Determines the low number for the scale.
High Range	Determines the high number for the scale.
Label Every	Determines the number of major ticks between labels.
Label Format	Allows user to determine how the label is displayed from 21 formats such as xxxx.
Text Color	Determines the color of the text.
Units Label	Determines the unit text for the labels.
Major Tick Divisions	Determines the number of ticks between major divisions.
Minor Tick Divisions	Determines the number of ticks between minor divisions.
Tick Color	Determines the color of the ticks.
Tick Style	Determines the style of the ticks: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge.
Scale Position	Determines the location of the scale relative to the slider track: left or bottom, right or top, or both.

Editing Background Parameters

To edit background parameters:

1. Double-click on the GWXSlider ActiveX control. The GWXSlider ActiveX Properties dialog box appears.
2. Select the Background tab (Figure 17-9).

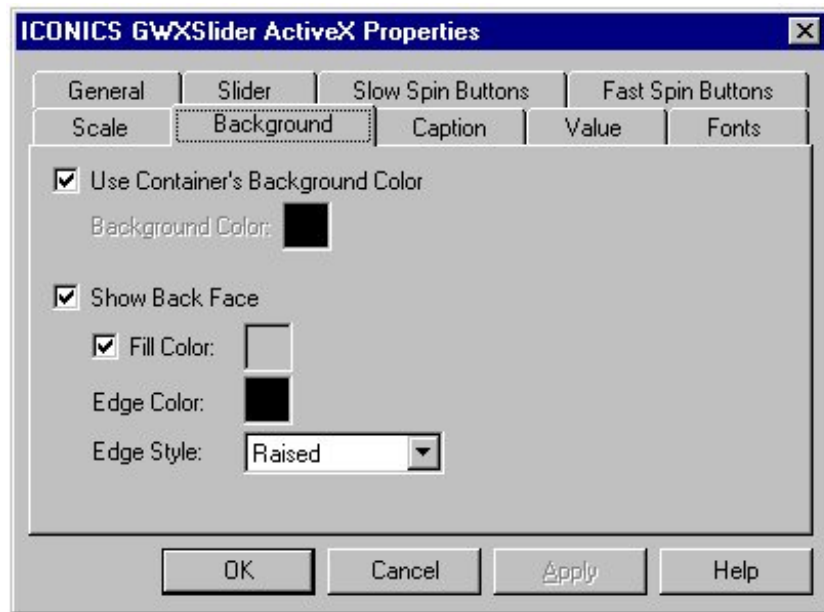


Figure 17-9: GWXSlider ActiveX Properties Dialog Box: Background Tab

3. Modify the parameters using Table 17-8.

Table 17-8: Background Tab Parameters

Parameter	Description
Use Container's Background Color	If checked, the background color of the slider matches the background color of the container. If unchecked, allows user to select a color for the background of the slider, which has a border around it.
Show Back Face	If checked, the slider has an edge border.
Fill Color	Allows user to select a color for the slider. Default is gray.
Edge Color	Allows user to select a color for the edge of the slider.
Edge Style	Allows user to select a style for the edge of the slider: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge. Default is raised.

Editing Caption Parameters

To edit caption parameters:

1. Double-click on the GWXSlider ActiveX control. The GWXSlider ActiveX Properties dialog box appears.
2. Select the Caption tab (Figure 17-10).

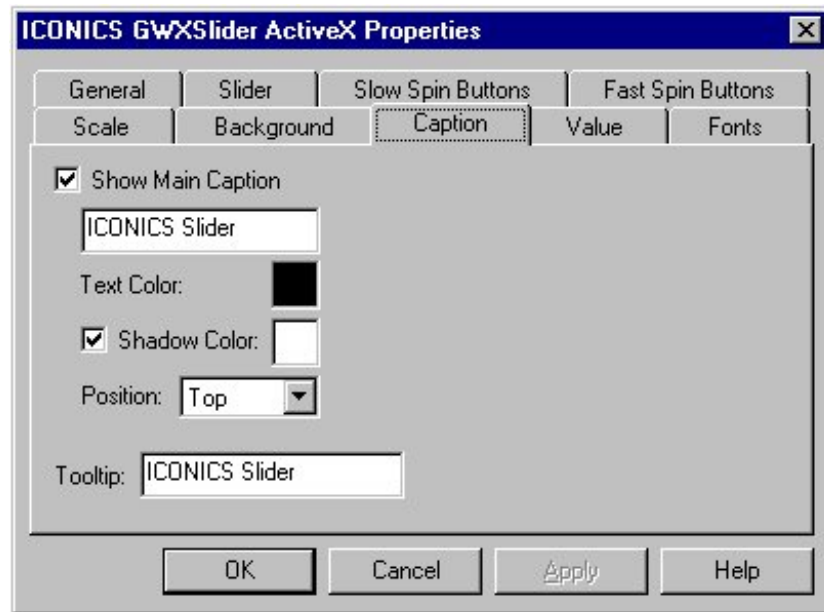


Figure 17-10: GWXSlider ActiveX Properties Dialog Box: Caption Tab

3. Modify the parameters using Table 17-9.

Table 17-9: Caption Tab Parameters

Parameter	Description
Show Main Caption	If checked, the text entered underneath appears as the title of the slider.
Text Color	Allows user to select a color for the text of the slider.
Shadow Color	If checked, allows user to select a color for the shadow of the caption text.
Position	Determines the position of the caption: left, right, top, or bottom.
ToolTip	Allows user to enter a word, phrase, or message to appear as a ToolTip to the user.

Editing Value Parameters

To edit value parameters:

1. Double-click on the GWXSlider ActiveX control. The GWXSlider ActiveX Properties dialog box appears.
2. Select the Value tab (Figure 17-11).

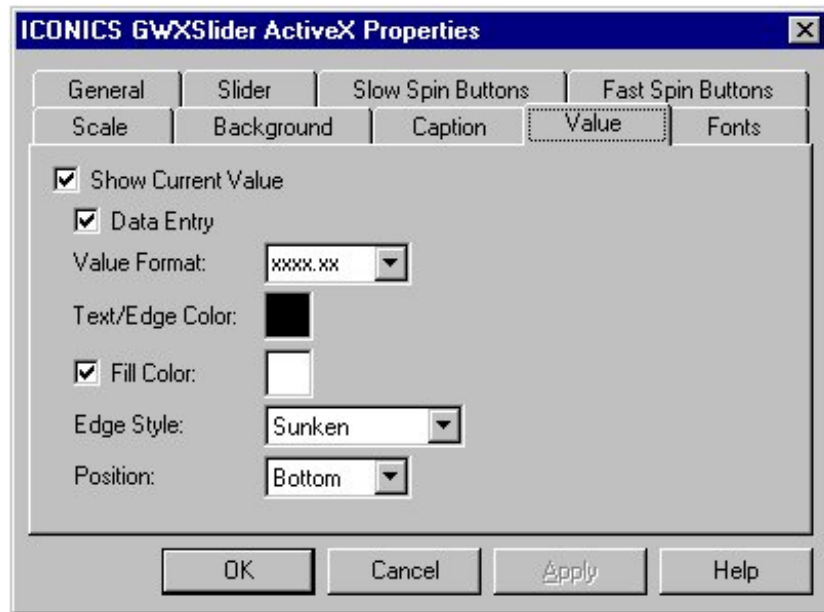


Figure 17-11: GWXSlider ActiveX Properties Dialog Box: Value Tab

3. Modify the parameters using Table 17-10.

Table 17-10: Value Tab Parameters

Parameter	Description
Show Current Value	If checked, the current value of the tag displays in a box at the bottom of the slider.
Data Entry	If checked, user may type in new data values into the box when in Runtime mode.
Value Format	Allows user to determine how the data is displayed from 21 numeric formats such as xxxx.
Text/Edge Color	Determines the color of the text and the edge of the box.
Fill Color	Determines the color of the box that displays the values.
Edge Style	Allows user to select a style for the edge of the box: thin line, medium line, thick line, etched, bump, raised, sunken, or no edge
Position	Determines the position of the value field relative to the slider track: left, right, top, or bottom.

Editing Fonts Parameters

To edit fonts parameters:

1. Double-click on the GWXSlider ActiveX control. The GWXSlider ActiveX Properties dialog box appears.
2. Select the Fonts tab (Figure 17-12).

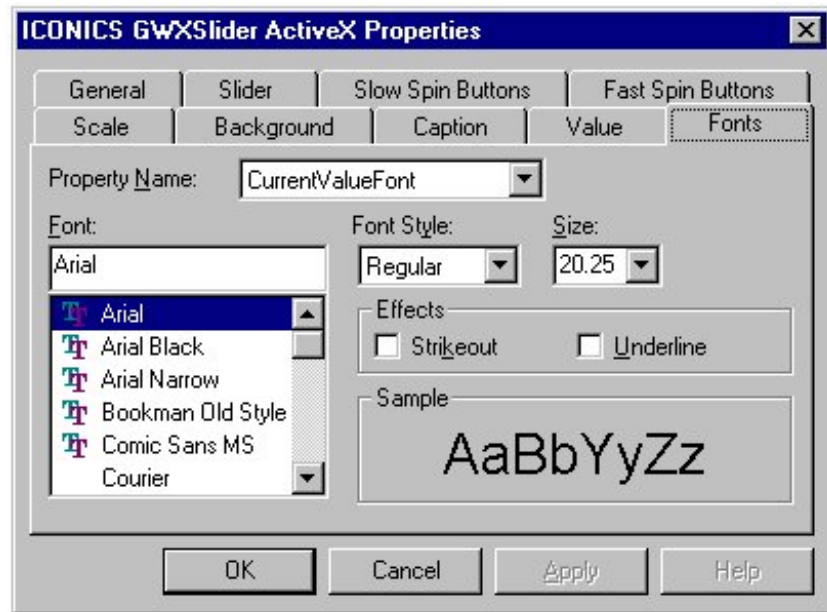


Figure 17-12: GWXSlider ActiveX Properties Dialog Box: Fonts Tab

3. Modify the parameters using Table 17-11.

Table 17-11: Fonts Tab Parameters

Parameter	Description
Property Name	Allows user to choose which text to change the font of: CurrentValueFont, MainCaptionFont, or ScaleLabelFont.
Font	Displays the current font and a scrolling list of fonts available.
Font Style	Determines the style of the font: regular, bold, italic, or bold italic.
Size	Allows user to select the size of the font from a drop-down menu.
Effects	Allows strikeouts and underlining of the text chosen.
Sample	Displays a real-size example of the font, style, and size chosen.

Viewing the About Information

To view the about information:

1. Double-click on the GWXSlider ActiveX control. The GWXSlider ActiveX Properties dialog box appears (Figure 17-13).

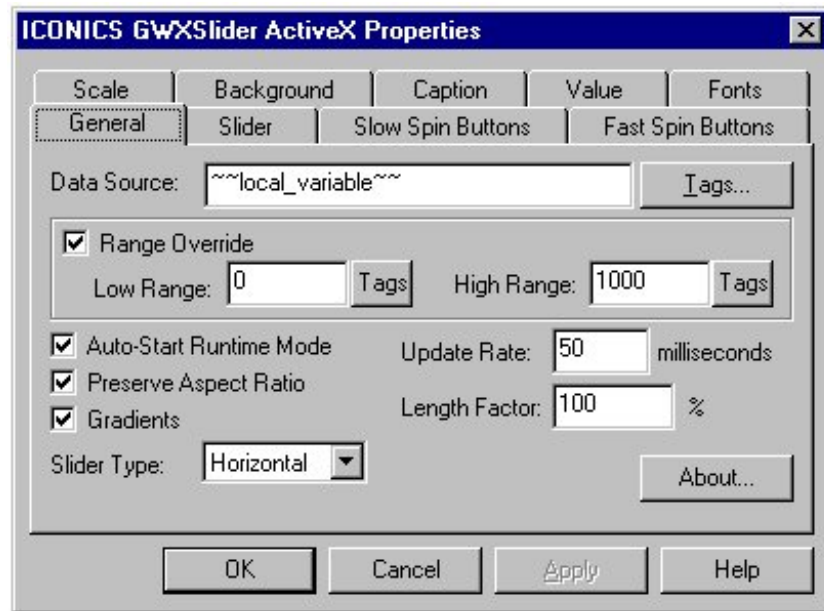


Figure 17-13: GWXSlider ActiveX Properties Dialog Box: General Tab

2. Select the General tab.
3. Select the About button. The Iconics About Box appears (Figure 17-14).

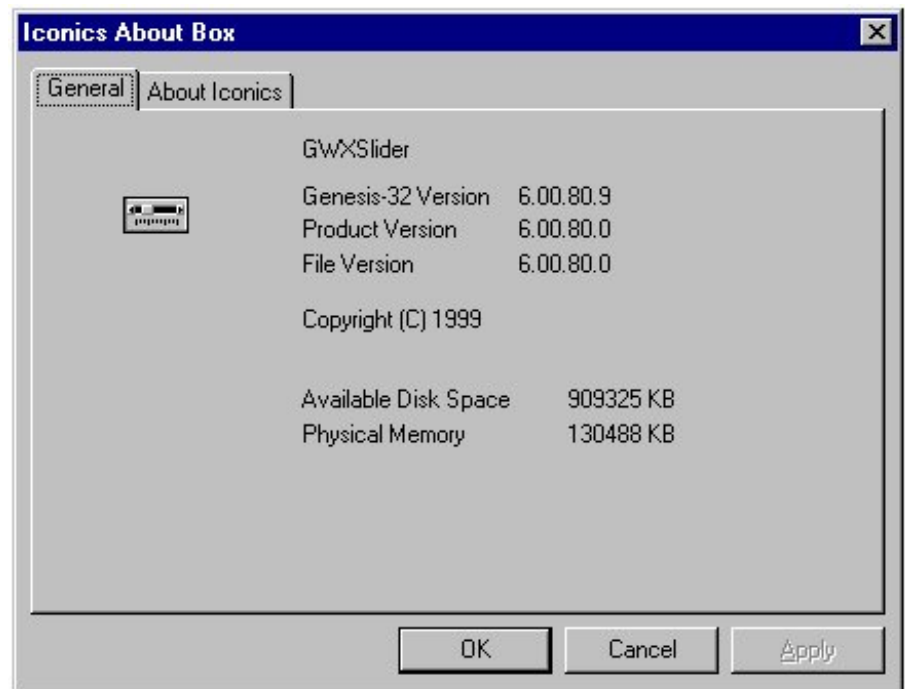


Figure 17-14: Iconics About Box

4. Select the General tab. The information listed includes version, copyright, etc.

Chapter 18

AWXView32 ActiveX® Control

Introduction

The M-Alarm Current Events Viewer is an ActiveX™ control. Insert the M-Alarm Current Events Viewer into any ActiveX container including M-Graphics and M3 Workstation. The AWXView32 ActiveX control provides real-time alarm detection, sorting, filtering, viewing, and reporting from OLE for Process Controls (OPC) Alarm and Event (AE) Servers. It is an M-Alarm Current Events Viewer, which means it only displays active alarms and current operator messages.

Note: Use the paste special function when copying the AlarmView32 ActiveX control.

The River of Time .ocx files are designed for use with only River of Time products.

Refer to *M-Alarm User's Guide* for more information.



Controls Group
507 E. Michigan Street
P.O. Box 423
Milwaukee, WI 53201

www.johnsoncontrols.com
Printed in U.S.A.

Appendix A

OLE Automation Reference

Introduction

OLE automation is used to access properties and methods of objects in M-Graphics.

Key Concepts

Properties and Methods

Properties are used to reference attributes if an object (for example, M-Graphics display) has a *BackgroundColor* property. Methods are used to make the object perform an action (for example, a display has a *FileOpen* method, which loads a new display).

Properties and methods can be called from Visual Basic scripts or from C++ programs. The user can access the methods and properties of M-Graphics objects to create new customized behavior for those objects.

There are numerous object types in M-Graphics each with its own methods and properties. This chapter describes, in detail, the methods and properties for the object types exposed by M-Graphics.

Custom Configuration

Every tab in the Property Inspector has a Custom button. The Pick Dynamic action, Launch Application brings up the OWSExec function for Metasys commands. The user programs the custom configuration utility, which is tied to the button. Clicking this button creates an Automation object, which has a certain program ID (ProgID). By default, the ProgID is IcoCustom.Configure. This ProgID may be overridden by setting an alternative ProgID in the Windows registry key. Expose the Automation object from a DLL. M-Graphics uses this dispatch interface of this object to call the method (Table A-1):

```
boolean CustomConfigure(long hWndParent, IDispatch* ObjectOne,
BSTR ObjectOneType, IDispatch* ObjectTwo, BSTR
ObjectTwoType)
```

Table A-1: Custom Configuration Definitions

Object	Description
hWndParent	Parent window handle of Property Inspector.
ObjectOne	Dispatch pointer of the M-Graphics object associated with the Gwx32 property page shown.
ObjectOneType	String name of the object type for ObjectOne. Note: Check the object type to verify the support configuration.
ObjectTwo	Dispatch pointer of the M-Graphics display where object belongs.
ObjectTwoType	String name of the object type for ObjectTwo.

The Return Value is True if changes were made to the objects passed to the specific method. The Return Value is False if no changes were made.

Custom Command Execution

This is one of the options for the Pick Dynamic action in M-Graphics. Run custom functions and applications using the Custom Command option. The custom application must be specified as either an executable (.EXE) or a DLL. Expose the Automation object from an .EXE. The .EXE allows multiple instances of this Automation object without launching multiple instances of the application.

The first time a Pick action is selected as a Custom Command, M-Graphics creates an Automation object, which has a certain ProgID. The parameters are the same as Custom Configure. **Do not** release the dispatch pointers passed to this method. After the method returns, M-Graphics does not release the dispatch pointer for the Automation object. M-Graphics keeps the dispatch pointer for the next time a Custom Command is executed. The dispatch pointer is not released until this instance of M-Graphics is closed.

GwxDisplay

GwxDisplay is the object type for M-Graphics displays. Properties and methods of the GwxDisplay object type are described in detail below. Also described are the events exposed to M-Graphics's integrated Visual Basic for Applications (VBA).

Properties

OLE_COLOR BackgroundColor - Gets/sets the current display's background color.

1. **Boolean Redraw** - When True, automation calls when changing the visual appearance of a display will automatically refresh the display. When False, the user must explicitly refresh the display. (For instance, you may want to change the attributes of many objects and refresh them all at once.)
2. **Short ScaleMode** - Sets the scaling mode of the current display. Valid values are:
 - FixedScale = 0
 - Scaleable = 1
 - ScaleablePreserveAspect = 2

Methods

1. **BSTR GetFileName()** - Returns the name of the currently loaded display.
2. **Boolean FileOpen(BSTR filename)** - Loads the specified display. Returns True for success; False for failure.
3. **Boolean FileSave()** - Saves the current display. Returns True for success; False for failure.
4. **Boolean FileSaveAs(BSTR filename)** - Saves the current display using the specified file name. Returns True for success; False for failure.
5. **Boolean FileNew()** - Starts a new (empty) display. Returns True for success; False for failure.
6. **Boolean FilePrint()** - Prints the current display. Returns True for success; False for failure.
7. **void ShowWindow()** - Shows the M-Graphics main window.
8. **void HideWindow()** - Hides the M-Graphics main window.
9. **void RefreshWindow()** - Redraws the visible portion of an M-Graphics display.
10. **void BringWindowToTop()** - Brings the M-Graphics main window to the top of the window z-order.
11. **void ExitApplication()** - Exits M-Graphics.
12. **void MinimizeWindow()** - Minimizes the M-Graphics main window.
13. **void MaximizeWindow()** - Maximizes the M-Graphics main window.
14. **void RestoreWindow()** - Restores the (non-minimized/non-maximized) window size and position.
15. **void SetWindowDimensionsPixels(long left, long top, long width, long height)** - Sets the M-Graphics main window size and location in pixels.
16. **void GetWindowDimensionsPixels(long* left, long* top, long* width, long* height)** - Gets the M-Graphics main window size and location in pixels.
17. **void SetWindowDimensionsPercent(float left, float top, float width, float height)** - Sets the M-Graphics main window size and location as a percentage of the total screen size. Parameters should be values in the range of 0.0 to 1.0.

18. **void GetWindowDimensionsPercent(float* left, float* top, float* width, float* height)** - Gets the M-Graphics main window size and location as a percentage of the total screen size. Retrieved parameters will have values in the range of 0.0 to 1.0.
19. **void GetDisplayDimensions(long* width, long* height)** - Retrieves the display dimensions (work area/world bounds) of the currently loaded display.
20. **void SetDisplayDimensions(long width, long height)** - Sets the display dimensions (work area/world bounds) of the currently loaded display.
21. **void SetViewDimensions(long left, long top, long width, long height)** - Sets the M-Graphics view rectangle's size and location. The view dimensions define what portion of the work area/whole display is visible. Setting the view dimensions can be used to zoom and pan the view of a display.
22. **void GetViewDimensions(long* left, long* top, long* width, long* height)** - Gets the M-Graphics view rectangle's size and location.
23. **void GetClientDimensionsPixels(long* left, long* top, long* width, long* height)** - Gets the M-Graphics client rectangle's size and location in pixels. The client rectangle is the area of the main window not including the borders, title bar, and menu bar.
24. **IDispatch* OpenPopupWindow(BSTR filename, Boolean modal, Boolean center, Boolean hidden)** - Opens an M-Graphics popup window. Returns the new popup window's object (of type GWXview32) on success; Null on failure.
25. **void CloseAllPopupWindows()** - Closes all currently open popup windows.
26. **Boolean ClosePopupWithTitle(BSTR titleSubstring)** - Closes the popup window which has the specified substring as part or all of the text in the popup window's title bar. Returns True if the specified popup window was successfully closed; False otherwise.
27. **IDispatch* GetOpenPopupWithTitle(BSTR titleSubstring)** - Gets the open popup window's object which has the specified substring as part or all of the text in the popup window's title bar. The popup window's object is of type GWXview32. Returns Null if no matching popup window is found.
28. **Boolean IsRuntimeMode()** - Returns True if M-Graphics is currently in Runtime mode, False otherwise.
29. **void StartRuntime()** - Puts M-Graphics into Runtime mode.

- 30. **void StopRuntime()** - Takes M-Graphics out of Runtime mode (into Configure mode).
- 31. **Boolean ToggleRuntime()** - Toggles the current operator mode between Configure mode and Runtime mode. Returns True, if the method put M-Graphics into Runtime mode; False if the method put M-Graphics into Configure mode.
- 32. **IDispatch* GetVisibleObjectFromName(BSTR objectName)** - Gets the visible object with the specified object name. The returned object type will be GwxVisible or one of GwxVisible derived object types (GwxRectangle, GwxEllipse, GwxLine, GwxText, GwxArc, GwxBitmap, GwxMetafile, GwxSymbol, GwxOleObject, GwxButton). Returns Null if no matching object is found.
- 33. **long GetNumberOfTopLevelVisibles()** - Returns the number of visible objects in the root symbol of a display (referred to as “Top Level Visibles,” because they are the objects in top-most level of the symbol hierarchy).
- 34. **IDispatch* GetVisibleObjectFromIndex(long index)** - Returns the Top Level Visible with the given zero-based index. An object with an index of zero is the object furthest back in the z-order. This function is useful for iterating through all the top level visible objects.
- 35. **Boolean PopCurrentSymbol()** - Pops the current symbol edit level up one level. Returns True for success; False for failure. (See also *GwxSymbol::PushCurrentSymbol()*). *Available only in Configure mode.*
- 36. **Boolean PopAllCurrentSymbol()** - Pops the current symbol edit level back to the root level. Returns True for success; False for failure. (See also *GwxSymbol::PushCurrentSymbol()*). *Available only in Configure mode.*
- 37. **IDispatch* GetDynamicObjectFromName(BSTR objectName)** - Gets the dynamic object with the specified object name. The returned object is of type GwxDynamic or one of the GwxDynamic derived types (GwxSize, GwxLocation, GwxRotation, GwxHide, GwxFlash, GwxPick, GwxDigitalColor, GwxAnalogColor, GwxAnimator, GwxAnalogSelector, GwxDigitalSelector, GwxProcessPoint, GwxTimedate). Returns Null if no matching object is found.
- 38. **IDispatch* GetPointObjectFromName(BSTR pointName)** - Gets the data point object with the specified point name. The returned object is of type GwxPoint. Returns Null if no matching object is found.

39. **Boolean DeleteObject(BSTR objectName)** - Deletes the visible object with the specified object name. Returns True for success; False for failure. *Available only in Configure mode.*
40. **Boolean DeleteDynamic(BSTR objectName)** - Deletes the dynamic object with the specified object name. Returns True for success; False for failure. *Available only in Configure mode.*
41. **void DeselectAllObjects()** - Deselects all currently selected visible objects. *Available only in Configure mode.*
42. **IDispatch* CreateEllipse(float left, float top, float width, float height, Boolean isFilled, OLE_COLOR fillColor, OLE_COLOR lineColor, long lineWidth, GWXLINESTYLE lineStyle, Boolean hasShadow, OLE_COLOR shadowColor, GWX3DEDGESTYLE edgeStyle, Boolean isHidden, BSTR objectName)** - Creates and returns GwxEllipse object with the specified attributes, in the current display. (See also *GwxEllipse* and *GwxVisible*.) *Available only in Configure mode.*
43. **IDispatch* CreateRectangle(float left, float top, float width, float height, Boolean isFilled, OLE_COLOR fillColor, OLE_COLOR lineColor, long lineWidth, GWXLINESTYLE lineStyle, Boolean hasShadow, OLE_COLOR shadowColor, GWX3DEDGESTYLE edgeStyle, Boolean isHidden, BSTR objectName, Boolean rounded)** - Creates and returns a GwxRectangle object with the specified attributes in the current display. (See also *GwxRectangle* and *GwxVisible*.) *Available only in Configure mode.*
44. **IDispatch* CreateText(float x, float y, BSTR text, long alignment, Boolean stretchText, Boolean isFilled, OLE_COLOR fillColor, OLE_COLOR lineColor, long lineWidth, GWXLINESTYLE lineStyle, Boolean hasShadow, OLE_COLOR shadowColor, GWX3DEDGESTYLE edgeStyle, Boolean isHidden, BSTR objectName)** - Creates and returns a *GwxText* object with the specified attributes in the current display. (See also *GwxText* and *GwxVisible*.) *Available only in Configure mode.*

- 45. **IDispatch* CreatePolyline(VARIANT vertices, Boolean isFilled, OLE_COLOR fillColor, OLE_COLOR lineColor, long lineWidth, GWXLINESTYLE lineStyle, Boolean hasShadow, OLE_COLOR shadowColor, GWX3DEDGESTYLE edgeStyle, Boolean isHidden, BSTR objectName)** - Creates and returns a GwxLine object with the specified attributes in the current display. The parameter vertices are an array of float values such that the Elements 0, 2, 4, 6... of the array are x-coordinates of the vertices of the line, and Elements 1, 3, 5, 7... of the array are y-coordinates of the vertices of the line. (See also *GwxLine* and *GwxVisible*.) *Available only in Configure mode.*
- 46. **IDispatch* CreateArc(GWXARCTYPE arcType, float centerX, float centerY, float radiusX, float radiusY, float startAngle, float endAngle, Boolean isFilled, OLE_COLOR fillColor, OLE_COLOR lineColor, long lineWidth, GWXLINESTYLE lineStyle, Boolean hasShadow, OLE_COLOR shadowColor, GWX3DEDGESTYLE edgeStyle, Boolean isHidden, BSTR objectName)** - Creates and returns a GwxArc object with the specified attributes, in the current display. Angles are specified in degrees. (See also *GwxArc* and *GwxVisible*.) *Available only in Configure mode.*
- 47. **IDispatch* CreateSymbol(BSTR objectName)** - Groups all currently selected objects into a GwxSymbol object giving the resulting symbol object the specified object name. Returns the resulting GwxSymbol object. (See also *GwxSymbol* and *GwxVisible*.) *Available only in Configure mode.*
- 48. **Boolean UngroupSymbol(BSTR objectName)** - Ungroups the GwxSymbol object with the specified object name. Although the function destroys the specified *GwxSymbol*, it does not destroy the objects that were grouped in that symbol. Returns True for success; False for failure. *Available only in Configure mode.*
- 49. **void DuplicateSelection()** - Creates duplicates of the currently selected objects. *Available only in Configure mode.*
- 50. **IDispatch* GetHeadObject()** - Gets the head visible object. In Configure mode, the head object is the selected object with the solid blue or red resize grips. In Runtime mode, the head object is the pickable object that currently has focus. If there is no head object, this method returns Null.

51. **Boolean SetHeadObject(BSTR objectName)** - Sets the head object to the visible object with the specified name. Returns True for success; False for failure.
52. **Boolean SetAliasDefinition(BSTR aliasName, BSTR newDefinition)** - Sets the alias definition of the specified alias name for all dynamic objects in the display. This function can be used in Runtime mode to easily change the data connections of dynamic objects on-the-fly. Returns False if no matching alias names were found; True otherwise. (See also *GwxVisible::SetAliasDefinition* and *GwxDynamic::SetAliasDefinition*).
53. **Boolean OpenTagBrowser(long hWndParent, BSTR* tagName)** - Opens the Tag Browser with the specified window handle as the parent window. The parameter tagName receives the tag name selected by the user. Returns True if the user presses the OK button; False if the user presses the Cancel button.
54. **Boolean SelectPaletteColor(OLE_COLOR* SelectedColor)** - Opens the M-Graphics Color Palette dialog box. The parameter SelectedColor receives the color selected by the user. Returns True if the user presses the OK button; False if the user presses the Cancel button.
55. **Boolean QueryRanges(BSTR dataSource, double* lowRange, double* highRange)** - Queries the OPC server for the high and low range values of the specified tag name (dataSource).
56. **Boolean QueryDataType(BSTR dataSource, GWXDATATYPE* dataType)** - Queries the OPC server for the data type of the specified tag name (dataSource). See *GwxPoint* for possible values of GWXDATATYPE.

57. **IDispatch* CreateSizeDynamic(BSTR visibleObjectName, BSTR dynamicObjectName, GWXSIZEDYNTYPE sizeType, Boolean clip, float startSize, float endSize)** - Creates a *GwxSize* object with the specified attributes and attaches it to the visible object with the specified object name. The parameters *startSize* and *endSize* are percentage values in the range of 0.0 to 1.0. Valid values for *GWXSIZEDYNTYPE* are:

- *SizeLeft* = 0
- *SizeRight* = 1
- *SizeUp* = 2
- *SizeDown* = 3
- *SizeUpLeft* = 4
- *SizeUpRight* = 5
- *SizeDownLeft* = 6
- *SizeDownRight* = 7
- *SizeLeftRight* = 8
- *SizeUpDown* = 9
- *SizeLeftRightBias* = 10
- *SizeUpDownBias* = 11
- *SizeAllFour* = 12
- *SizeLeftRightUp* = 13
- *SizeLeftRightDown* = 14
- *SizeUpDownLeft* = 15
- *SizeUpDownRight* = 16

Returns the newly created *GwxSize* object on success; Null if the operation failed. (See also *GwxSize* and *GwxDynamic*.) *Available only in Configure mode.*

58. **IDispatch* CreateLocationDynamic (BSTR visibleObjectName, BSTR dynamicObjectName, float offsetX, float offsetY, Boolean slider, Boolean tracking, short numberOfDetents, Boolean continuousUpdate)** - Creates a GwxLocation object with the specified attributes and attaches it to the visible object with the specified object name. The parameters offsetX and offsetY refer to the distance the object will travel from its current location. Returns the newly created GwxLocation object on success; Null if the operation failed. (See also *GwxLocation* and *GwxDynamic*.) *Available only in Configure mode.*
59. **IDispatch* CreateRotationDynamic(BSTR visibleObjectName, BSTR dynamicObjectName, float startAngle, float endAngle, float pivotX, float pivotY, Boolean clockwise, Boolean dial, Boolean tracking, short numberOfDetents, Boolean continuousUpdate)** - Creates a GwxRotation object with the specified attributes and attaches it to the visible object with the specified object name. The parameters pivotX and pivotY are offsets from the center of the object. Angles are specified in degrees. Returns the newly created GwxRotation object on success; Null if the operation failed. (See also *GwxRotation* and *GwxDynamic*.) *Available only in Configure mode.*
60. **IDispatch* CreateHideDynamic(BSTR visibleObjectName, BSTR dynamicObjectName, Boolean hideWhenTrue, Boolean disableObject)** - Creates a GwxHide object with the specified attributes and attaches it to the visible object with the specified object name. Returns the newly created GwxHide object on success; Null if the operation failed. (See also *GwxHide* and *GwxDynamic*.) *Available only in Configure mode.*
61. **IDispatch* CreateFlashDynamic(BSTR visibleObjectName, BSTR dynamicObjectName, Boolean hideObject, Boolean flashWhenTrue, Boolean altStateWhenOff, Boolean changeFill, Boolean changeLine, Boolean changeShadow, OLE_COLOR altFillColor, OLE_COLOR altLineColor, OLE_COLOR altShadowColor)** - Creates a GwxFlash object with the specified attributes and attaches it to the visible object with the specified object name. Returns the newly created GwxFlash object on success; Null if the operation failed. (See also *GwxFlash* and *GwxDynamic*.) *Available only in Configure mode.*

62. **IDispatch* CreateAnalogColorDynamic(BSTR visibleObjectName, BSTR dynamicObjectName, Boolean changeFill, Boolean changeLine, Boolean changeShadow, OLE_COLOR startFillColor, OLE_COLOR endFillColor, OLE_COLOR startLineColor, OLE_COLOR endLineColor, OLE_COLOR startShadowColor, OLE_COLOR endShadowColor, Boolean defaultColorAbove, Boolean defaultColorBelow)** - Creates a *GwxAnalogColor* object with the specified attributes and attaches it to the visible object with the specified object name. Returns the newly created *GwxAnalogColor* object on success; Null if the operation failed. (See also *GwxAnalogColor* and *GwxDynamic*.) *Available only in Configure mode.*
63. **IDispatch* CreateDigitalColorDynamic(BSTR visibleObjectName, BSTR dynamicObjectName, Boolean changeColorWhenTrue, Boolean changeFill, Boolean changeLine, Boolean changeShadow, long fillColor, long lineColor, long shadowColor, BSTR dataSource)** - Creates a *GwxDigitalColor* object with the specified attributes and attaches it to the visible object with the specified object name. The object is created with one initial data connection. Additional data connections can be added by calling this function again for the same visible object. Returns the newly created *GwxDigitalColor* object on success; Null if the operation failed. (See also *GwxDigitalColor*, *GwxDigitalColorInfo*, and *GwxDynamic*.) *Available only in Configure mode.*

64. **IDispatch* CreatePickDynamic(BSTR visibleObjectName, BSTR dynamicObjectName, GWXPICKACTION pickAction, GWXBUTTONTYPE pickType, GWXEXECUTIONTRIGGER executionTrigger, GWXMOUSEBUTTON mouseButton, Boolean initiallySelected, BSTR groupName, BSTR fileName, Boolean modal, Boolean center, BSTR value1, BSTR value2, BSTR value3)** - Creates a GwxPick object with the specified attributes and attaches it to the visible object with the specified object name. Valid values for GWXPICKACTION are:

- PickLoadDisplay = 0
- PickDragDropLoad = 1
- PickPopupWindow = 2
- PickDownloadValue = 3
- PickToggleValue = 4
- PickLaunchApp = 5
- PickClose = 6
- PickRunScript = 7

Valid values for GWXBUTTONTYPE are:

- ButtonNormal = 0
- ButtonCheck = 1
- ButtonRadio = 2

Valid values for GWXEXECUTIONTRIGGER are:

- TriggerOnDown = 1
- TriggerWhileDown = 2
- TriggerOnDnWhileDn = 3
- TriggerOnUp = 4
- TriggerOnDnOnUp = 5
- TriggerWhileDnOnUp = 6
- TriggerOnDnWhileDnOnUp = 7

Valid values for GWXMOUSEBUTTON are:

- MouseButtonLeft = 0
- MouseButtonMiddle = 1
- MouseButtonRight = 2

Depending upon the value of `pickAction`, some parameters may be ignored by M-Graphics. Returns the newly created `GwxPick` object on success; Null if the operation failed. (See also *GwxPick* and *GwxDynamic*.) *Available only in Configure mode.*

65. **IDispatch* CreateButton(long buttonType, float x, float y, BSTR label, GWXTEXTALIGNMENT alignment, Boolean stretchText, Boolean isFilled, long fillColor, long lineColor, long lineWidth, long lineStyle, Boolean hasShadow, long shadowColor, long edgeStyle, Boolean isHidden, BSTR objectName)** - Creates and returns a `GwxButton` object with the specified attributes in the current display. A button object will be non-operational until a `GwxPick` object is attached to it. (See also *GwxButton*, *GwxPick*, *GwxDynamic*, *GwxText*, and *GwxVisible*.) *Available only in Configure mode.*
66. **IDispatch* CreateTimedate(BSTR textObjectName, BSTR dynamicObjectName, long formatType, BSTR timeFormat, BSTR dateFormat)** - Creates a `GwxTimedate` object with the specified attributes and attaches it to the `GwxText` object with the specified object name. Returns the newly created `GwxTimedate` object on success; Null if the operation failed. (See also *GwxTimedate*, *GwxText*, *GwxVisible*, and *GwxDynamic*.) *Available only in Configure mode.*
67. **IDispatch* CreateProcessPoint(BSTR textObjectName, BSTR dynamicObjectName, GWXDATATYPE dataType, Boolean update, Boolean dataEntry, Boolean hasInitialValue, VARIANT initialValue, BSTR format)** - Creates a `GwxProcessPoint` object with the specified attributes and attaches it to the `GwxText` object with the specified object name. Returns the newly created `GwxProcessPoint` object on success; Null if the operation failed. (See also *GwxProcessPoint*, *GwxText*, *GwxVisible*, and *GwxDynamic*.) *Available only in Configure mode.*
68. **IDispatch* CreateAnimator(BSTR symbolName, BSTR dynamicObjectName, Boolean animateWhenTrue, Boolean visibleWhenOff, Boolean currentFrameWhenOff)** - Creates a `GwxAnimator` object with the specified attributes and attaches it to the `GwxSymbol` object with the specified object name. Returns the newly created `GwxAnimator` object on success; Null if the operation failed. (See also *GwxAnimator*, *GwxSymbol*, *GwxVisible*, and *GwxDynamic*.) *Available only in Configure mode.*

69. **IDispatch* CreateAnalogSelector(BSTR symbolName, BSTR dynamicObjectName, Boolean hiddenWhenAbove, Boolean hiddenWhenBelow)** - Creates a GwxAnalogSelector object with the specified attributes and attaches it to the GwxSymbol object with the specified object name. Returns the newly created GwxAnalogSelector object on success; Null if the operation failed. (See also *GwxAnalogSelector*, *GwxSymbol*, *GwxVisible*, and *GwxDynamic*.) *Available only in Configure mode.*
70. **IDispatch* CreateDigitalSelector(BSTR symbolName, BSTR dynamicObjectName)** - Creates a GwxDigitalSelector object with the specified attributes and attaches it to the GwxSymbol object with the specified object name. Returns the newly created GwxDigitalSelector object on success; Null if the operation failed. The newly created object has no data connections; to add data connections, use *GwxDigitalSelector::SetConnectionInfo*. (See also *GwxDigitalSelector*, *GwxDigitalSelectorInfo*, *GwxSymbol*, *GwxVisible*, and *GwxDynamic*.) *Available only in Configure mode.*

Events

This section describes the events that are exposed to M-Graphics's integrated Visual Basic for Applications scripting.

1. **void DisplayLoad()** - This event is fired just after a display is loaded into M-Graphics.
2. **void DisplayUnload()** - This event is fired just before a display is unloaded from M-Graphics (i.e., just prior to loading a new display into M-Graphics or prior to closing M-Graphics).
3. **void PreRuntimeStart()** - This event is fired just before M-Graphics is put into Runtime mode.
4. **void PostRuntimeStart()** - This event is fired when M-Graphics has finished entering Runtime mode.
5. **void PreRuntimeStop()** - This event is fired just before M-Graphics exits Runtime mode.
6. **void PostRuntimeStop()** - This event is fired after M-Graphics has exited Runtime mode.
7. **void PreAnimateDisplay()** - This event is fired just before M-Graphics animates a display (animating a display is the process of requesting the tags from the OPC servers). Displays are animated when loaded during Runtime mode and deanimated when unloaded during Runtime mode.
8. **void PostAnimateDisplay()** - This event is fired after M-Graphics has finished animating a display.

9. **void PreDeanimateDisplay()** - This event is fired just before M-Graphics deanimates a display (deanimating a display is the process of releasing the tags previously requested from the OPC servers).
10. **void PostDeanimateDisplay()** - This event is fired after M-Graphics has finished deanimating a display.
11. **void MouseDown(long Button, long Shift, float X, float Y)** - This event is fired when a mouse button is clicked within the M-Graphics window. Values for Button are:
 - Left button is pressed = 1
 - Right button is pressed = 2
 - Middle button is pressed = 4Values for Shift are:
 - No Key is pressed = 0
 - Shift key is pressed = 1
 - Ctrl key is pressed = 2
 - Alt key is pressed = 4The parameters X and Y indicate where the mouse was clicked. These coordinates are relative to the upper left corner of the window.
12. **void MouseUp(long Button, long Shift, float X, float Y)** - This event is fired when a mouse button is released.
13. **void DblClick(long Button, long Shift, float X, float Y)** - This event is fired when a mouse button is double clicked within the M-Graphics window.
14. **void MouseMove(long Button, long Shift, float X, float Y)** - This event is fired when the mouse is moved within the M-Graphics window. The parameter Button is zero if no mouse buttons are currently pressed during the move.
15. **void KeyDown(long KeyCode, long Shift)** - This event is fired when a keyboard key is pressed while the M-Graphics window has focus.
16. **void KeyUp(long KeyCode, long Shift)** - This event is fired when a keyboard key is released.

GwxVisible

GwxVisible is the object type from which visible M-Graphics objects (ellipses, rectangles, etc.) are derived. In other words, all visible M-Graphics objects have the properties and methods of **GwxVisible**.

Properties

1. **Boolean Visible** - True if the object is visible, False if the object is hidden.
2. **OLE_COLOR FillColor** - Gets/sets the visible object's fill color.
3. **Boolean IsFilled** - True if the object is filled, False if the object is not filled.
4. **OLE_COLOR LineColor** - Gets/sets the visible object's line/border color.
5. **OLE_COLOR ShadowColor** - Gets/sets the visible object's shadow color.
6. **Boolean HasShadow** - True if the object has a shadow, False if the object does not have a shadow.
7. **long LineWidth** - The width of an object's line/border. This value must be in the range of 0 to 10.
8. **GWXLINESTYLE LineStyle** - The style of the object's line/border. Valid values for GWXLINESTYLE are:
 - LineSolid = 0
 - LineDash = 1
 - LineDot = 2
 - LineDashDot = 3
 - LineDashDotDot = 4
 - LineNone = 5
9. **GWX3DEDGESTYLE EdgeStyle** - The 3-D edge style of the object's border. Valid values for GWX3DEDGESTYLE are:
 - EdgeNone = 0
 - EdgeRaised = 5
 - EdgeEtched = 6
 - EdgeBump = 9
 - EdgeSunken = 10

10. **Boolean Selected** - True if the object is selected; False if the object is not selected. Selected objects are objects that have resize grips around them. This property can only be changed in Configure mode.
11. **float Angle** - Gets/sets the rotation angle of the object. The angle is specified in degrees.
12. **BSTR ObjectName** - Gets/sets the object name of a visible object. The object name is used to identify the object when using certain OLE Automation methods (for instance, `GwxDisplay::GetVisibleObjectFromName`). M-Graphics will ensure that object names are unique. If you assign an object name that already exists for another visible object in a display, M-Graphics will append a number to the object name (for example, "tank" would become "tank1", "tank1" would become "tank2," etc.).
13. **BSTR UserDescription** - A description string for the visible object. Typically, this string is displayed as informational text in a ToolTip.
14. **BSTR UserCustomData** - This string is used to store custom data. Use this property to associate any additional data with the visible object.

Methods

1. **void MoveObject(float offsetx, float offsety)** - Moves the object by the specified offset. This method only works for objects that are top level visible objects.
2. **void RefreshObject()** - Causes the visible object to be redrawn.
3. **void SetObjectDimensions(float left, float top, float width, float height)** - Sets the object's size and location. This method only works for objects that are top level visible.
4. **void GetObjectDimensions(float* left, float* top, float* width, float* height)** - Gets the object's size and location.
5. **void StretchObject(float scaleX, float scaleY, float anchorX, float anchorY)** - Stretches the object based on the specified scale factors. The parameters, `scaleX` and `scaleY`, are values such that 0.0 is 0% and 1.0 is 100%. The anchor values are used to change the way the object shifts position during stretching. For example, to resize an object from its center, the anchor values should be the center coordinates of the object. This method only works for objects that are top level visible.

6. **Boolean SetAliasDefinition(BSTR aliasName, BSTR newDefinition)** - Sets the alias definition for all dynamic objects associated with this visible object. If this GwxVisible is also a GwxSymbol, this function is recursively applied to all objects grouped within the symbol. The function returns False if no matching alias names were found; True otherwise. (See also *GwxDisplay::SetAliasDefinition* and *GwxDynamic::SetAliasDefinition*).
7. **IDispatch* GetDynamicObjectFromName(BSTR nameSubstring)** - Gets the dynamic object with the specified name (or portion of a name), which is attached to this visible object.
8. **IDispatch* GetDynamicObjectFromIndex(long index)** - Gets the dynamic object with the specified zero based index which is attached to this visible object (index zero is the first attached dynamic). This function is useful for iterating through all the dynamic objects attached to this visible object.
9. **long GetNumberOfDynamics()** - Returns the number of dynamic objects attached to this visible object.

GwxText

GwxText has all the properties and methods of **GwxVisible** plus the additional properties described below.

Properties

1. **BSTR Text** - This string is the text that is displayed by the text object.
2. **Boolean StretchText** - True if the font size should be resized when the text object is stretched. False if the font size should remain the same when the object is stretched.
3. **GWXTEXTALIGNMENT Alignment** - The alignment of the text (left, center, right). This property is only significant for multiline text strings. Valid values for **GWXTEXTALIGNMENT** are:
 - TextAlignLeft = 0
 - TextAlignCenter = 1
 - TextAlignRight = 2
4. **IFontDisp* Font** - Gets/sets font of the text object.

GwxRectangle

GwxRectangle has all the properties and methods of **GwxVisible** plus the additional properties described below.

Properties

1. **Boolean Rounded** - When getting this property, if the value is True, the rectangle has rounded corners. When setting this property to True, the rectangle is given rounded corners with the default rounding settings.
2. **long RoundingX** - Gets/sets the amount of horizontal rounding of the rectangle's corners.
3. **long RoundingY** - Gets/sets the amount of vertical rounding of the rectangle's corners.

GwxArc

GwxArc has all the properties and methods of **GwxVisible** plus the additional properties described below.

Properties

1. **GWXARCTYPE ArcType** - Gets/sets the arc type of the arc. Valid values for **GWXARCTYPE** are:
 - **ArcArc** = 0
 - **ArcPie** = 1
 - **ArcChord** = 2
2. **float StartAngle** - Gets/sets the start angle of the arc (in degrees).
3. **float EndAngle** - Gets/sets the end angle of the arc (in degrees).

GwxSymbol

GwxSymbol has all the properties and methods of **GwxVisible** plus the additional methods described below.

Methods

1. **Boolean PushCurrentSymbol()** - Pushes this symbol object onto the symbol edit stack. This essentially makes the objects grouped in this symbol top level visible objects. (Some operations, like **GwxVisible::SetObjectDimensions**, are only allowed on top level objects.)
2. **IDispatch* GetVisibleObjectFromName(BSTR nameSubstring)** - Gets the visible object with the specified name (or portion of a name), which is an immediate child of this symbol.
3. **IDispatch* GetVisibleObjectFromIndex(long index)** - Gets the visible object with the specified zero based index which is an immediate child of this symbol. (Index zero is the first child object.) This function is useful for iterating through all the immediate child objects grouped in this symbol.
4. **long GetNumberOfChildVisibles()** - Returns the number of immediate child objects grouped in this symbol.

GwxOleObject

GwxOleObject has all the properties and methods of **GwxVisible** plus the additional method described below.

Method

IDispatch* GetOLEObject() - Gets the actual OLE object wrapped by **GwxOleObject**.

GwxEllipse

GwxEllipse has all the properties and methods of **GwxVisible**.

GwxLine

GwxLine has all the properties and methods of **GwxVisible**.

GwxBitmap

GwxBitmap has all the properties and methods of **GwxVisible**.

GwxMetafile

GwxMetafile has all the properties and methods of **GwxVisible**.

GwxButton

GwxButton has all the properties and methods of **GwxVisible** and **GwxText**.

GwxDynamic

GwxDynamic is the object type from which dynamic M-Graphics objects (size dynamic, location dynamic, etc.) are derived. In other words, all dynamic M-Graphics objects have the properties and methods of **GwxDynamic**.

Properties

1. **BSSTR DataSource** - The primary data source for the dynamic object. DataSource is a string, which represents an OPC tag name, an expression, a constant value, or an M-Graphics local variable.
2. **Boolean RangeOverride** - Gets/sets range override status. When False, M-Graphics will use the ranges associated with the primary DataSource. When True, M-Graphics will use the ranges defined in the HighRange and LowRange properties. This property is only used for dynamics based on an analog data source (GwxSize, GwxLocation, GwxRotation, GwxAnalogColor, GwxAnalogSelector, GwxProcessPoint); it is ignored for dynamics that are based on digital connections.
3. **BSSTR HighRange** - Represents the overridden high range for this dynamic. HighRange is a string, which represents an OPC tag name, an expression, a constant value, or an M-Graphics local variable. This property is only used for dynamics based on an analog data source; it is ignored for dynamics that are based on digital connections.
4. **BSSTR LowRange** - Represents the overridden low range for this dynamic. LowRange is a string, which represents an OPC tag name, an expression, a constant value, or an M-Graphics local variable. This property is only used for dynamics based on an analog data source; it is ignored for dynamics that are based on digital connections.
5. **long TimerRate** - Frequency update rate for timer based dynamic types. (This property is ignored for dynamics that are not timer based.) Timer based dynamics include: GwxFlash, GwxAnimator, and GwxPick.

6. **BSTR ObjectName** - Gets/Sets the object name of a dynamic object. The object name is used to identify the object when using certain OLE Automation methods (for instance, *GwxDisplay::GetDynamicObjectFromName*). M-Graphics will ensure that object names are unique. If you assign an object name that already exists for another dynamic object in a display, M-Graphics will append a number to the object name (for example, "size" would become "size1," "size1" would become "size2," etc.).
7. **BSTR UserDescription** - A description string for the dynamic object. Typically, this string is displayed as informational text in a ToolTip.
8. **BSTR UserCustomData** - This string is used to store custom data. Use this property to associate any additional data with the dynamic object.

Methods

1. **Boolean SetAliasDefinition(BSTR aliasName, BSTR newDefinition)** - Sets the alias definition for this dynamic object only. Returns False if no matching alias names were found, True otherwise. (See also *GwxDisplay::SetAliasDefinition* and *GwxVisible::SetAliasDefinition*.)
2. **IDispatch* GetVisibleObject()** - Gets the visible object to which this dynamic is attached.

GwxDigitalSelector

GwxDigitalSelector has all the properties and methods of **GwxDynamic** plus the additional methods described below.

Methods

Boolean SetConnectionInfo(short objectNumber, BSTR dataSource, Boolean showWhenTrue) - Sets data connections for this digital selector object. The parameter objectNumber is a zero based index representing an immediate child of this symbol object to which this dynamic is attached (index zero is the first child object). (See also *GwxDigitalSelectorInfo*.)

GwxDigitalSelectorInfo

Currently unused. In the future versions of M-Graphics, this object may include properties of individual data connections for **GwxDigitalSelector** objects.

GwxDigitalColor

GwxDigitalColor has all the properties and methods of **GwxDynamic**.

GwxDigitalColorInfo

Currently unused. In the future versions of M-Graphics, this object may include properties of individual data connections for **GwxDigitalColor** objects.

GwxAnalogSelector

GwxAnalogSelector has all the properties and methods of **GwxDynamic**.

GwxAnalogColor

GwxAnalogColor has all the properties and methods of **GwxDynamic**.

GwxAnimator

GwxAnimator has all the properties and methods of **GwxDynamic**.

GwxSize

GwxSize has all the properties and methods of **GwxDynamic**.

GwxLocation

GwxLocation has all the properties and methods of **GwxDynamic**.

GwxRotation

GwxRotation has all the properties and methods of **GwxDynamic**.

GwxFlash

GwxFlash has all the properties and methods of **GwxDynamic**.

GwxHide

GwxHide has all the properties and methods of **GwxDynamic**.

GwxPick

GwxPick has all the properties and methods of **GwxDynamic**.

GwxProcessPoint

GwxProcessPoint has all the properties and methods of **GwxDynamic**.

GwxTimedate

GwxTimedate has all the properties and methods of **GwxDynamic**.

GwxPoint

GwxPoint is the object type that M-Graphics uses to represent data connections. If several GwxDynamic objects are connected to the same data source, they reference a single shared GwxPoint object. GwxPoint objects handle OPC tags, expressions, constant values, and M-Graphics local variables. GwxPoint objects cannot be explicitly created or destroyed. M-Graphics automatically manages the lifetimes of GwxPoint objects based on the data source connections of the GwxDynamic objects in the display.

Properties

1. **VARIANT Value** - Current data value of this point object. This property gets updated with new values during Runtime mode.
2. **VARIANT HighRange** - High range value associated with this point object.
3. **VARIANT LowRange** - Low range value associated with this point object.
4. **GWXDATATYPE DataType** - Data type of this point object. Valid values for GWXDATATYPE are:
 - `DataTypeShort` = 2
 - `DataTypeLong` = 3
 - `DataTypeFloat` = 4
 - `DataTypeDouble` = 5
 - `DataTypeString` = 8
 - `DataTypeBool` = 11
 - `DataTypeByte` = 17

Methods

BSTR GetPointName() - Gets the point name (data source string) of this point object.

GWXview32

GWXview32 is the object type for the M-Graphics ActiveX control (GWXview32.ocx) and for M-Graphics popup windows.

Properties

1. **short BorderStyle** - Border style of the window. Valid values are:
 - No border = 0
 - Normal Border = 1
2. **short Appearance** - Appearance of the window border. Valid values are:
 - Flat = 0
 - 3-D = 1
3. **BSTR DisplayName** - File name of the M-Graphics display loaded in this object/control. Setting this property to a new file name will cause the new display to be loaded.
4. **Boolean UseAmbientBackColor** - When True, the background color of the object/control will automatically be set to match the background color of the container in which this object is embedded. This property only works if the container supports the AmbientBackColor property.
5. **Boolean AutoStartRuntime** - When True, the object/control will automatically enter Runtime mode when the object's container enters Runtime mode. This property only works if the container supports the AmbientUserMode property.
6. **Boolean OverrideScrollbarSettings** - When True, this property indicates the object/control will override the scroll bar visibility settings of the display currently loaded in the control. When False, the object/control will use the scroll bar settings defined in the currently loaded display.
7. **Boolean VerticalScrollbar** - When True, the vertical scroll bar of the control's window is visible; when False the vertical scroll bar is hidden. This property is ignored if **OverrideScrollbarSettings** is False.
8. **Boolean HorizontalScrollbar** - When True, the horizontal scroll bar of the control's window is visible; when False the horizontal scroll bar is hidden. This property is ignored if **OverrideScrollbarSettings** is False.

Methods

1. **void StartRuntime()** - Puts this object/control into Runtime mode.
2. **void StopRuntime()** - Takes this object/control out of Runtime mode.
3. **IDispatch* GetDisplay()** - Returns the GwxDisplay object for the display currently loaded in this object/control. You can then use this object to access all of the properties and methods of all the object types described in this chapter.

Appendix B

DIN Symbol Library

Introduction

This chapter provides examples of the DIN symbols included with M-Graphics. Symbols are continually added; refer to the software for a complete set.

Key Concepts

Symbol Library

The symbol libraries with prebound tags provided with M-Graphics, aid in the creation of facility drawings. Users can just modify the alias name to change the binding reference.

There are four categories of symbols: DIN, French, Johnson Controls, and other.

To import a symbol in M-Graphics, click the Import Symbol icon



or on the Draw menu, select Import and click on Symbol.

DIN Symbols Examples

Table B-1: DIN Symbols Examples






































































































					
621	623	625	627	641	643
					
647	181	182	183	184	185
					
191	192	193	194	195	196
					
101	102	103	104	105	107
					
116	231	232	233	234	235
					
141	142	143	144	145	146
					
171	172	173	174	175	176
					
121	122	125	252	253	

Table B-1: DIN Symbols Examples (Cont.)

					
138	221	222	254	211	212
					
201	202	203	204	205	206
					
501	502	503	504	511	512
					
521	523	525	526	527	528
					
529	530	531	532	533	534
					
536	537	541	542	544	545
					
546	547	548	550	551	630
					
637	638	652	653	655	656
					
581	582	583	584	587	588

Appendix C

French Symbol Library

Introduction

This chapter provides examples of the French symbols included with M-Graphics. Symbols are continually added; refer to the software for a complete set.

Key Concepts

Symbol Library

The symbol libraries with prebound tags provided with M-Graphics, aid in the creation of facility drawings. Users can just modify the alias name to change the binding reference.

There are four categories of symbols: DIN, French, Johnson Controls, and other.

To import a symbol in M-Graphics, click the Import Symbol icon



or on the Draw menu, select Import and click on Symbol.

French Symbols Examples

Table C-1: French Symbols Examples

































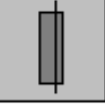














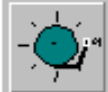

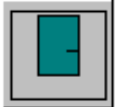
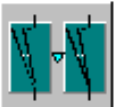

















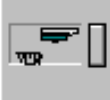

					
access	dooracc1	hornalm1	proxrdr1	radar1	reader1
					
chlalm1	chlrst1	dryclal1	dryclfn1	cooling battery	heating battery
					
adjust	start/stop	switch1	FireDamper1	d&v pos.	damper1
					
damper2	detect1	fire	fire dampers	fzondet1	smoke ex
					
sprinkler1	sprinklers	contact1	contalm1	contdif1	contpos1
					
conts1	fcontact1	fuse1	trans1	sfalm1	sfstat1
					
filter1	btry1	elebtry1	elegen1	ups1	clocks
					
hardware	plumbing	htg	tempex1	Humid1	buzzalm1

Table C-1: French Symbols Examples (Cont.)

					
detectr1	dooracc1	invalm1	light1	airflo1	dtherm1
					
gaine1	hum1	pipe1	pressure1	pressw1	stherm1
					
elec	swcont1	swpos1	valve2c1	valve2h1	valve3c1
					
valve3h1	vlvpos	vcr	video1		

Appendix D

Johnson Controls Symbol Library

Introduction

This chapter provides examples of the Johnson Controls symbols included with M-Graphics. Symbols are continually added; refer to the software for a complete set.

Key Concepts

Symbol Library

The symbol libraries with prebound tags provided with M-Graphics, aid in the creation of facility drawings. Users can just modify the alias name to change the binding reference.

There are four categories of symbols: DIN, French, Johnson Controls, and other.

To import a symbol in M-Graphics, click the Import Symbol icon



or on the Draw menu, select Import and click on Symbol.

Johnson Controls Symbols Examples

Table D-1: Johnson Controls Symbols Examples


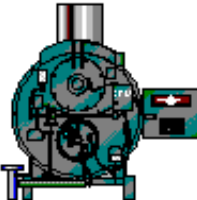
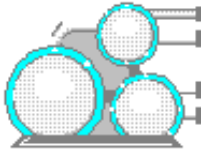
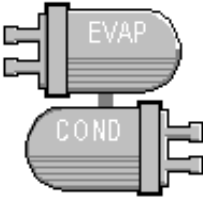
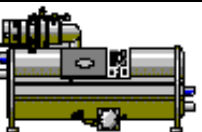
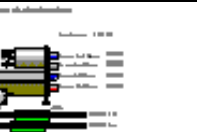
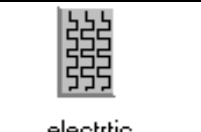
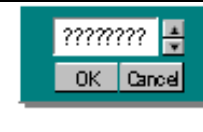
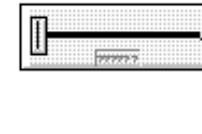

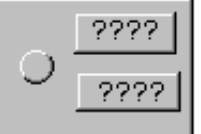
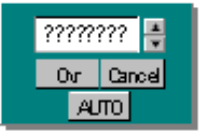

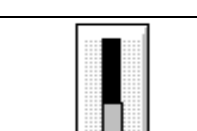
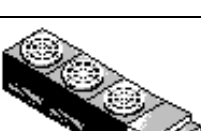


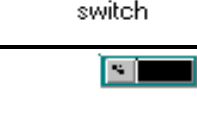
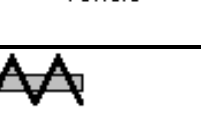





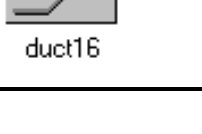
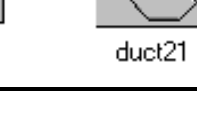
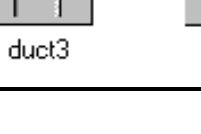
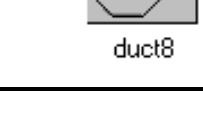




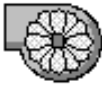
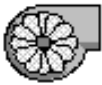


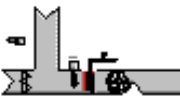
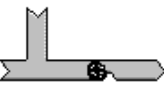
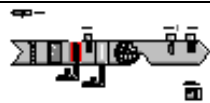

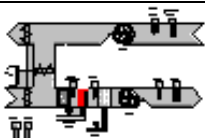
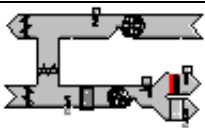
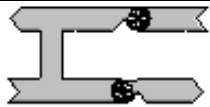
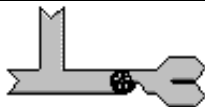
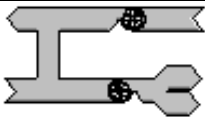
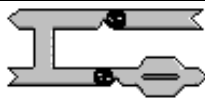


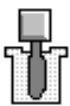

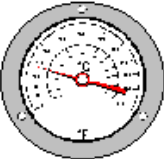

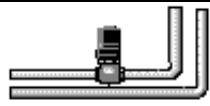

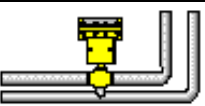














			
M9100	actuator1	d&v pos.	damp1
			
AirStr	Airflow	Arrow1	Arrow4
			
duct16	duct17	duct21	duct3
			
duct1	duct13	duct6	duct8
			
start/stop	switch	Tower3	Chillwtr1
			
adjustSlider	Tower1	cmdr0verride	override
			
cw coil	electric	hw coil	adjust
			
Chill1	Chill2	EVAP	COND

Table D-1: Johnson Controls Symbols Examples (Cont.)

					
Return Fan	Supply Fan	Filt1	H1	H2	S2
					
H3	H4	H5	H7		
					
S1	S4	S5	S6		
					
HeatEx	Hum-1	Temp5	kilowatt	temp	volt
					
2-Way	2-Way	3-Way	Pitot4	Temp2	
					
Humid1	Pitot1	Pitot2	Press1	Press3	Press6
					
Temp3	Temp1	Temp10	Temp11	Temp4	Temp8

Appendix E

Other Symbol Library

Introduction

This chapter provides examples of the Other symbols included with M-Graphics. Symbols are continually added; refer to the software for a complete set.

Key Concepts

Symbol Library

The symbol libraries with prebound tags provided with M-Graphics, aid in the creation of facility drawings. Users can just modify the alias name to change the binding reference.

There are four categories of symbols: DIN, French, Johnson Controls, and other.

To import a symbol in M-Graphics, click the Import Symbol icon



or on the Draw menu, select Import and click on Symbol.

Other Symbols Examples

Table E-1: Other Symbols Examples




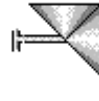


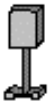













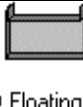



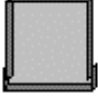
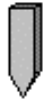


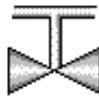




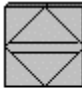



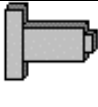








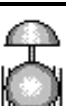

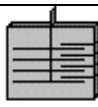













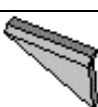















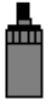
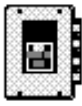









					
3-D 3-way solenoid valve with reset	3-D 3-way valve	3-D 4-way solenoid valve	3-D Angle blowdown valve	3-D Adsorption column	3-D Adsorption dryer
					
3-D Agitator	3-D Agitator	3-D Bag filter	3-D Ball valve	3-D Belt conveyor	3-D Belt dryer
					
Interbus	3-D Blowdown valve	3-D Circuit breaker	3-D Column	3-D Cone roof tank	3-D Control valve with hand actuator
					
3-D Drum dryer	3-D Evaporator	3-D Floating roof tank	3-D Flow nozzle	3-D Flow regulator, self-contained	3-D Flow straightening vane
					
3-D Gas holder	3-D Hopper	3-D Gate valve	3-D Gyratory crusher	3-D Hand control valve	3-D Helical mixer
					
3-D Packed column	3-D Horizontal vessel	3-D Plug valve	3-D Programmable logic control, auxiliary location	3-D Restriction orifice drilled in valve	3-D Rotary separator
					
3-D Ribbon blender	3-D Rod/ball mix	3-D Roll blender	3-D Roll crusher	3-D Roll stand	3-D Rotary actuated

Table E-1: Other Symbols Examples (Cont.)

					
3-D Rotary calciner	3-D Rotary dryer	Thermometer	Motor 1	3-D Rotary plug valve	3-D Screw conveyor
					
3-D Rotary shelf dryer	3-D Scrubber	3-D Settler thickener	3-D Solid bowl centrifuge	3-D Sphere	3-D Spray dryer
					
Rotary plug valve	3-D Target type sensor	3-D Trap	RF capacitance level meter	3-D Static mixer	3-D Stop check valve
					
3-D Vacuum safety valve	3-D Valve	3-D Vibrating screen	3-D Solid bowl centrifuge	PID Loop Controller 1	3-D Vacuum crystallizer
					
3-D Variable area flow indicator	3-D Weigh hopper	3-D pachuca crystallizer	120° straight (long)	180° angle 1	Check valve - horizontal
					
Control valve - vertical	Cracked pipe - horizontal	Cracked cut-away	Flange with bolts on left	Flow element	Hand valve
					
Accelerometer	Gas detector	Cut-away 4	Elevated Tank 1	LED-style cut-away with 10 segments	Liquid storage drum
					
Storage facility 4	Control valve	Valve 11	Desktop PC	Flow meter	

Tips and Hints

Introduction

This appendix suggests ways to improve efficiency while using M-Graphics. This chapter describes how to:

- add graphic Hot Links to other screens
- add fan rotation to a graphic
- add command operation box to object when clicking left or right mouse button
- add focus window screen to object when clicking left or right mouse button
- create a slider bar
- add on/off binary commands to pushbuttons
- add override binary commands to pushbuttons
- add the auto binary command to a pushbutton

Key Concepts

Characteristics of Good Graphics

A good graphic serves one of the following two functions:

1. *To Inform*: A graphical representation of data (e.g., green for normal temperature values) and notifies the operator of certain data conditions (e.g., flashing condition when in alarm).
2. *To Navigate*: Provides quick and intuitive means to view various aspects of building equipment and areas (building pictures, floors, mechanic systems).

Benefits of good graphics:

- allows operators to quickly make decisions or evaluations
- demonstrates the advantages of graphics over text based information
- overcomes limitations of text based information
- upholds the different expectations of graphics vs. text based information
- upholds standards of graphics
- upholds standards of symbol usage, templates, and files
- estimated time to create graphics
- researches symbol examples from graphic library
- explains how much data to display on the screen
- explains navigational schemes
- explains existing standards that customers have in place
- saves money by helping to solve problems

Usable Graphics

The following list describes how to create a usable graphic:

1. Develop a standard to follow. The standard defined by Johnson Controls is suitable.
2. Use storyboarding to simplify and speed up the process.
3. Identify advantages of graphics:
 - a. Quickly see alarms and statuses due to color changes.
 - b. Physical locations of devices shown in piping, duct, or floor plans.
 - c. Data from different systems can be in one display (room temperatures all on one screen).
 - d. Relationships between equipment can be emphasized (boiler with pumps and valves shown with air handlers).
 - e. Utilize riser diagrams for network layout, AHU, or fire systems.
4. Identify limitations of text displays:
 - a. Data on the screen.
 - b. Operator must read each point value.
5. Identify user expectations from graphic:
 - a. Customize to suit the user.
 - b. Utilize photos or other previously created drawings.
 - c. Must come up fast.
 - d. Must contain required information.
6. Provide estimated time to create graphics:
 - a. First graphic takes the longest to create.
 - b. Duplicate a graphic for similar applications.
 - c. Check library for pre-drawn graphics.
7. Provide a minimum standard to follow:
 - a. Standards prevent guessing about fonts and styles.
 - b. Saves time.
 - c. Supports corporate standards.

8. Decipher and use standard symbols and files:
 - a. Symbols library- create, rename, delete, categories, and folders.
 - b. Examples library.
 - c. Drag-and-drop feature to create new standard symbols.
 - d. Naming of objects and shared key words.
 - e. Use templates to reduce duplications.
9. Decide what graphic types to use:
 - a. 2D, 3D, or Line.
 - b. DIN, French, Johnson Controls symbols, or other types.
10. Provide navigational schemes:
 - a. System Map could follow the Operator Workstation (OWS) Network Map.
 - b. Menu bar buttons.
 - c. Penetrate from building to floor plans to system types.
 - d. Capable to link to the Internet sites.
11. Allow data density on a graphic:
 - a. Various view areas.
 - b. Quantity of objects on the display.
12. Choose fonts, color palette, control buttons, and display standards.
13. Discuss: appearance, style, organization, navigation, controls, display, and templates.
14. Provide general rules and standards.

Note:	To ensure stability and provide steady performance to N1 devices, make sure the items included in the graphics are bound and connected. The total number of invalid (unbound) and unconnected (offline) items should be less than five percent, preferably none, of all included items in all graphics.
-------	---

Keyboard Shortcuts

Table F-1: Keyboard Shortcuts

Menu	Command	Shortcut
Edit	Undo	Ctrl+Z
	Cut	Ctrl+X
	Copy	Ctrl+C
	Paste	Ctrl+V
	Delete	Del
	Duplicate	Ctrl+D
	VBA Properties Window	F4
Space Evenly	Across	Ctrl+Shift+A
	Down	Ctrl+Shift+D
Make Same Size	Height	Ctrl+Shift+H
	Width	Ctrl+Shift+W
	Both (Height and Width)	Ctrl+Shift+O
View	Home	Ctrl+H
	Zoom 50%	Ctrl+1
	Zoom 75%	Ctrl+2
	Zoom 100%	Ctrl+3
	Zoom 150%	Ctrl+4
	Zoom 250%	Ctrl+5
	Custom	Ctrl+0
	Unzoom	Ctrl+Shift+U
	Zoom Selection	Ctrl+Shift+S
	Fit to Window	Ctrl+F
	Show Whole Display	Ctrl+W
	Toggle Toolbars	Ctrl+T
	Status Bar	Ctrl+B
	Toggle Both Scroll Bars	Ctrl+L
	Grid	Ctrl+Shift+G
Arrange	Group Into Symbol	Ctrl+G
	Ungroup Symbol	Ctrl+U
	Bring To Front	Ctrl+Shift+PgUp
	Send To Back	Ctrl+Shift+PgDn
	Bring Forward	Ctrl+PgUp
	Send Backward	Ctrl+PgDn
Align	Tops	Ctrl+Shift+T
	Bottoms	Ctrl+Shift+B
	Middles	Ctrl+Shift+M
	Lefts	Ctrl+Shift+L
	Rights	Ctrl+Shift+R
	Center	Ctrl+Shift+C
Configure	Runtime Mode	Ctrl+M

Additional M-Graphics Shortcuts and Features

The following are all additional M-Graphics Shortcuts and Features that help in creating and maintaining displays.

Arrow Key Resize

While in the Configuration mode, change the size and shape of the object using the arrow keys, shift key, and control key. Holding down the Control and Shift keys while pressing the arrow keys adjusts the bottom and left side of the graphic. Holding down the shift key and pressing the arrow keys adjusts the top right side of the object.

Tab Key Object Selection

Use the Tab key to move the selector from one object to another. The selector moves around the display in the order the objects were created. If the object were put in a particular order, the selector would move from the farthest object forward.

Graphic Libraries

The North American Graphics Standardization team has created a small library of sample graphics and symbols. The National Standard Graphics library contains pre-engineered sample graphics that can be a starting place for building graphics. Standard graphics can be added to *The Advisor* library at any time by uploading them to the M-Graphics page. For more information, refer to the *Advisor/Express/Exchange* Web sites. Information is located at The Advisor>Engineering/Technical section under Conferences and Field Support Center.

Procedure Overview

Table F-2: Tips and Hints

To Do This	Follow These Steps:
Add Graphic Hot Links to Other Screens	From the Command menu, select Dynamics > Intrinsic > Pushbutton. Left click on the graphic screen. From the Pick tab, select Action > Load Action. Using the Browse button, select the graphic file needed for the Hot Link. Select Open. From the Button tab, replace the highlighted button text with the Hot Link graphic name. Select OK.
Add Fan Rotation to a Graphic	Paste the fan on graphic. Right click fan. Select Edit Symbol. Right click on blade. Select Property Inspector. Select Rotation. Select Expression. Select Tags. Pick Object > Present_Value. Select Arithmetic > Multiplication. Select Shift plus Tags buttons together. Select Gwsim.ramp.long. Select OK. From the Rotation tab, select Range Override. Enter 0 for the Low Range. Enter 1000 for the High Range. Select OK and save.
Add Command Operation Box to Object when Clicking Left or Right Mouse Button*	Right click object. Select Dynamics > Action > Pick. Select the Pick tab. Select Action > Launch Application. Select the mouse button. Select Custom button. Select CMD from the Metasys Application. Select Browse button. Pick Object > Present_Value. Select OK. Delete Present_Value from OWSExec command line parameter. Select OK. Select OK and save file.
Add Focus Window Screen to Object when Clicking Left or Right Mouse Button*	Right click object. Select Dynamics > Action > Pick. Select Pick tab. Select Action > Launch Application. Select Left or Right mouse button. Select Custom. Select FOC from Metasys Application. Select Browse. Pick Object > Present_Value. Select OK. Delete Present_Value for the OWSExec command line parameter. Select OK. Select OK and save file.
Create a Slider Bar	Paste slider bar onto graphic. Right click slider bar. Select Edit Symbol. Right click Slider tab. Select Property Inspector. Select Location tab. Enter High and Low Range Override limits. Select Edit Aliases. Select Tag Browser box. Select Object > Present_Value. Select OK and save.
Add On/Off Binary Commands to Pushbuttons	From the Command menu, select Dynamics > Intrinsic > Pushbutton. Left click on the graphic screen. From the Pick tab, select Action > Download Value. Select Tags. Pick Object > Present_Value. Select OK. From the Pick tab, enter 1 or 0 into tags = entry box. From the Button tab, replace the highlighted button text with the state1 or state0 action.
Add Override Binary Commands to Pushbuttons*	From the Command menu, select Dynamics > Intrinsic > Pushbutton. Left click on graphic screen. From the Pick tab, select Action > Download Value. Select Tags. Pick Object > N1_Display_OV. Select OK. From the Pick tab, enter 1 or 0 into tags = entry box. Select OK.
Add the Auto Binary Command to a Pushbutton*	From the Command menu, select Dynamics > Intrinsic > Pushbutton. Left click on the graphic screen. From the Pick tab, select Action > Download Value. Select Tags. Pick Object > N1_Display_OV. Select OK. From the Pick tab, enter "\$Auto\$" into tags = entry box. From the Button tab, replace the highlighted button text with AUTO. Select OK.

*Note: Applies only to M5 Workstation.

Detailed Procedures

Adding Graphic Hot Links to Other Screens

To add graphic Hot Links to other screens:

1. From the Command menu, select Dynamics > Intrinsic > Pushbutton.
2. Left click on the graphic screen.
3. From the Pick tab, select Action > Load Action.
4. Using the Browse button, select the graphic file needed for the Hot Link.
5. Select Open.
6. From the Button tab, replace the highlighted button text with the Hot Link graphic name.
7. Select OK.

Adding Fan Rotation to a Graphic

To add a fan rotation to a graphic:

1. Paste the fan on graphic.
2. Right click fan.
3. Select Edit Symbol.
4. Right click on blade.
5. Select Property Inspector.
6. Select Rotation.
7. Select Expression.
8. Select Tags.
9. Pick Object > Present_Value.
10. Select Arithmetic > Multiplication.
11. Select Shift plus Tags Buttons together.
12. Select Gwsim.ramp.long.
13. Select OK.
14. From the Rotation tab, select Range Override.
15. Enter 0 for the Low Range.
16. Enter 1000 for the High Range.
17. Select OK and save.

Adding Command Operation Box to Object when Clicking Left or Right Mouse Button

To add command operation box to an object when clicking left or right mouse button:

1. Right click object.
2. Select Dynamics > Action > Pick.
3. Select the Pick tab.
4. Select Action > Launch Application.
5. Select mouse button (left/right).
6. Select Custom button.
7. Select CMD from Metasys Application.
8. Select Browse button.
9. Pick Object > Present_Value.
10. Select OK.
11. Delete Present_Value from OWSExec command line parameter.
12. Select OK.
13. Select OK and save file.

Note: This applies only to the M5 Workstation.

Adding Focus Window Screen to Object when Clicking Left or Right Mouse Button

To add focus window screen to object when clicking left or right mouse button:

1. Right click object.
2. Select Dynamics > Action > Pick.
3. Select Pick tab.
4. Select Action > Launch Application.
5. Select left or right mouse button.
6. Select Custom.
7. Select FOC from Metasys Application.
8. Select Browse. Pick Object > Present_Value.
9. Select OK.
10. Delete Present_Value from the OWSExec command line parameter.

11. Select OK.
12. Select OK and save file.

Creating a Slider Bar

To create a slider bar:

1. Paste slider bar onto graphic.
2. Right click slider bar.
3. Select Edit Symbol.
4. Right click Slider tab.
5. Select Property Inspector.
6. Select Location tab.
7. Enter High and Low Range Override limits.
8. Select Edit Aliases.
9. Select Tag Browser box.
10. Select Object > Present_Value.
11. Select OK and save.

Adding On/Off Binary Commands to Pushbuttons

To add on/off binary commands to pushbuttons:

1. From Command menu, select Dynamics > Intrinsic > Pushbutton.
2. Left click on the graphic screen.
3. From the Pick tab, select Action > Download Value.
4. Select Tags.
5. Pick Object > Present_Value.
6. Select OK.
7. From the Pick tab, enter 1 (for the object's state1 action) or 0 (for the object's state0 action) into tags = entry box.
8. From the Button tab, replace the highlighted button text with the state1 or state0 action (i.e., on or off).
9. Select OK.

Adding Override Binary Commands to Pushbuttons

To add override binary commands to pushbuttons:

1. From the Command menu, select Dynamics > Intrinsic > Pushbutton.
2. Left click on the graphic screen.
3. From the Pick tab, select Action > Download Value.
4. Select Tags.
5. Pick Object > N1_Display_OV.
6. Select OK.
7. From the Pick tab, enter 1 (object's state1 action) or 0 (object's state0 action) into tags = entry box.
8. Select OK.

Note: This applies only to the M5 Workstation.

Adding the Auto Binary Command to a Pushbutton

To add the auto binary command to a pushbutton:

1. From the Command menu, select Dynamics > Intrinsic > Pushbutton.
2. Left click on the graphic screen.
3. From the Pick tab, select Action > Download Value.
4. Select Tags.
5. Pick Object > N1_Display_OV.
6. Select OK.
7. From the Pick tab, enter "\$"AUTO"\$ into tags = entry box.
8. From Button tab, replace the highlighted button text with AUTO.
9. Select OK.

Note: This applies only to the M5 Workstation.

Appendix G

Guidelines and Recommendations for Better Graphics Performance with Integrated N1 Networks

Introduction

This chapter discusses guidelines and recommendations for designing an M-Graphics display for better performance with integrated N1 networks. This appendix describes how to:

- change the maximum update rate
- select multiple files and change the maximum update rate

Key Concepts

Guidelines and Recommendations for Better Graphics Performance Overview

M-Graphics Release 4.0 introduced the use of layers, which allows a drawing to have multiple layers. This feature enables each layer to have its own symbols, dynamics, and data. Use layers to simplify creating and displaying graphics.

To keep the M-Graphics response time acceptable and reduce the communication load on the N1 network, follow the guidelines in this section.

Target Maximum of 150 Objects Attributes Tagged per Graphic

In graphics that use more than 150 tags, distribute tags to hidden M-Graphics layers that have OLE for Process Controls (OPC) server requests disabled when hidden. (Refer to *Arranging Objects [LIT-6440070]*.)

However, the maximum number of tags, including all the graphics' layers, should not exceed a total of 150. Multiple tags that reference the same object, but call different attributes, have less impact on performance than tags that each reference a unique object.

Note: M-Web does not support most of the M-Graphics animation features, including the layering features. See the *Commissioning Metasys® M-Web Technical Bulletin (LIT-643210)* for more information.

Target Maximum of 300 Object Tags per Network Control Module (NCM) for a Starfield Display

On a single NCM, limit the number of object tags to 300 object tags per Starfield Display.

Note: If tags displayed in a Starfield Display are divided among multiple NCMs, the maximum is 1,000 object tags.

Target Maximum of 300 Objects Attributes/Tags Accessed per NCM at Any One Time, Across All Display Features and All Display Devices

This maximum includes logs, trends, graphics, and Metasys Data Visualization (MDV) features. Since this limit can be difficult to determine, consider adding diagnostic data points with alarm limits to detect if NCM idle time is consistently less than 50%. For more information on the Analog Data (AD) Object, refer to the *Reference* section of the *DDL Programmer's Manual (LIT-630040)*.

Each operator device requesting data has its own instance for data signup and loads the NCM more heavily, even if each device requests the same points. A single device requesting multiple instances of the same type of data does not load the NCM as heavily, because data requests are coordinated within the device.

Modifiable Parameters

To improve performance, modify the following parameters in the n1opc.ini file:

- Adjust the PollRate to values between 40,000 to 60,000 milliseconds (ms) (40 to 60 seconds) for jobs with more than 100 tags per graphics, if the Network Controller (NC) idle time is under 50% when graphical features run. (Expect slower analog value refresh in M-Graphics at these time values). Binary changes and analog alarm data are still reported whenever a change occurs. The default is 20,000 milliseconds (20 seconds).
- Set DormantCycles to between 2 and 10 instead of the default of 72 (72 gives 12-hour dormancy in the cache) for faster release of unused points from the OPC cache. For normal operations, a value of 6 is recommended (6 gives 1-hour dormancy).

Note: For more information regarding the n1opc.ini file, refer to *Appendix B: N1 OPC Data Access Server Advanced Concepts Technical Bulletin (LIT-643325)*.

Maximum Update Rate Default Value

The Maximum Update Rate determines how often data is obtained from the OPC server and from the Simulation Values. The Maximum Update Rate default value was 50 milliseconds for Release 1 through Release 5. This value is unnecessarily fast for the needs of our OPC servers. The default value has been increased to 1,000 milliseconds (1 second) to reduce the Central Processing Unit (CPU) activity. However, the use of 1,000 milliseconds is too slow for Simulated Values.

The Maximum Update Rate is specified in two locations for graphic files (.gdf) and template files (.tdf):

- for current display: Format > Display Properties
- for new displays: Format > Application Preferences

To reduce CPU activity, raise the Maximum Update Rate from 50 to 1,000 on all existing graphic and template files.

<p>Important: Symbols that utilize Simulation Values for animation no longer function properly when the maximum update rate is increased. These symbols may be replaced with symbols that use the Animator dynamic, which has its own update timer. The NA Standard Symbol Library has been revised to remove all Simulation Values.</p>

Refer to *Changing the Maximum Update Rate* in this chapter for more information.

Performance Tips

Special consideration is needed when using the Animator Dynamic. When developing graphic displays, ask yourself: How many animations are required? Would consolidation help? What rate are the graphics going to animate? The answers to questions, such as these, improve the performance of the graphic displays.

General Tips

The following tips help the users with displays:

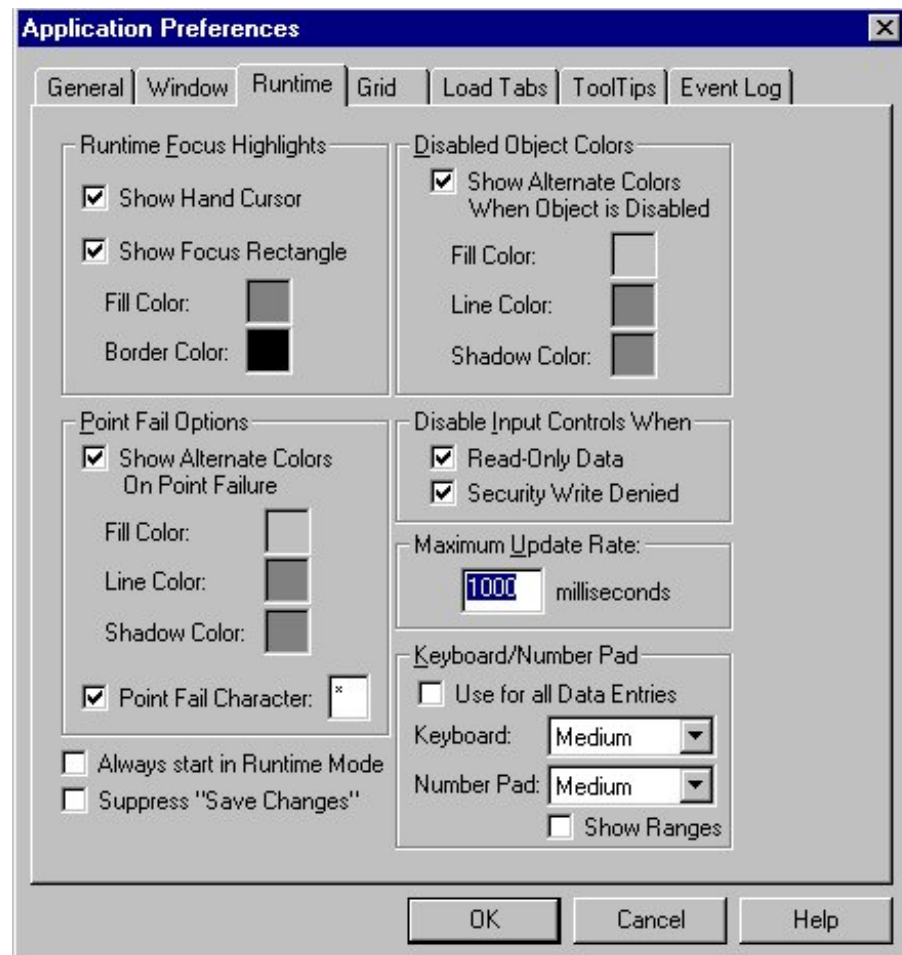
- Do not use full path names for file references. This property enables accessibility from other machines without the files' being in the same directory. For example, in the M-Graphics Load action, use *MyDisplay.gdf* instead of *C:\ProgramFiles\M-Graphics\JohnsonControls\MyProject.gdf*.
- Use Set Working Directory to establish a common location for file storage. To make updates easier, when working with a group of developers, have everyone point to a common location on the network.
- Use Shared Object Name property for objects in more than one place on more than one screen. This property allows all the files to be globally updated at the same time.
- Use Object Name property for every object. This property allows easier identification of objects when performing search/replace, Alias lists, and Find reports.
- Drag the file out of the M-Graphics display directly to the desktop of your computer, when sharing one object with someone else. This property creates a scrap file, which can be shipped over networks or copied to a floppy. Afterwards, drag the scrap file back into another M-Graphics display.
- Duplicate and place objects in one step by holding down the CTRL key while dragging.
- Open up multiple instances of M-Graphics to copy objects between different displays.
- Select two objects first instead of using two separate Hide dynamics to toggle the visibility. Then apply the Digital Selector dynamic. For more information, refer to the Gen32Demo.
- Use a sunken 3D Edge and fill color for Process Points: white, if it permits data entry; gray if it is read-only. This property allows operators to see whether a field is writable.
- Copy the master object and right-click on another object, then choose Paste Visible Properties from the pop-up menu to quickly give objects a similar look and feel (fill color, gradient, line style).
- Use templates to facilitate a common look and feel across an entire set of displays. When you update the entire project, only the template requires the updates.

Detailed Procedures

Changing the Maximum Update Rate

To change the Maximum Update Rate:

1. On the Format menu, select Application Preferences. The Application Preferences Dialog Box appears.
2. Select the Runtime tab (Figure G-1).



**Figure G-1: Application Preferences Dialog Box:
Runtime Tab**

3. Change the Maximum Update Rate to the new value in milliseconds.

Note: Depending on the graphic type, changing the value affects the dynamics of the graphic.

4. Click OK.

Selecting Multiple Files and Changing the Maximum Update Rate

To select multiple files and change the Maximum Update Rate:

1. On the Format menu, select Apply Preferences to Properties. The Choose Settings Dialog Box appears (Figure G-2).

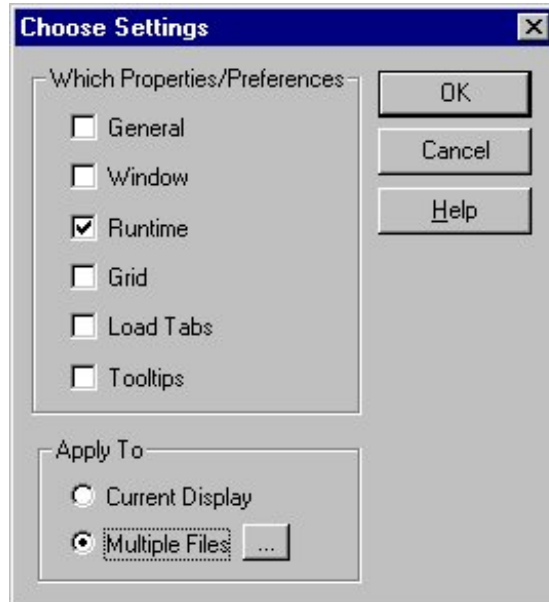



Figure G-2: Choose Settings Dialog Box

Note: Uncheck all properties except Runtime to reduce the risk of applying properties to all files.

2. Select Multiple Files.

3. Click the  button to add desired .gdf files. The Select Files Dialog Box appears (Figure G-3).

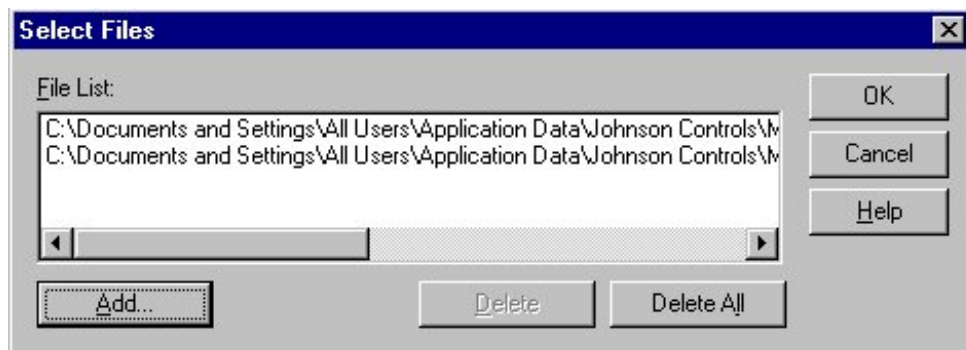


Figure G-3: Select Files Dialog Box

4. Click Add and browse to desired files.
5. Click OK to add the desired files. The Select Files Dialog Box closes.
6. Click OK.

Note: Set the Maximum Update Rate in all newly created
M-Graphics files.



Controls Group
507 E. Michigan Street
P.O. Box 423
Milwaukee, WI 53201

www.johnsoncontrols.com
Printed in U.S.A.